InterviewAI: Real-Time Questions Generator Using LLM

BY

SOON CHUN HONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JANUARY 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Dr. Aun Yichiet, for his exceptional guidance, unwavering support, and insightful advice throughout the course of my Final Year Project (FYP2). His vast expertise in the field of artificial intelligence has been instrumental in shaping the direction and success of my project, named InterviewAI. From the outset, Dr. Aun Yichiet encouraged me to push the boundaries of my knowledge and abilities, helping me navigate the complexities of AI with confidence and clarity. His dedication to my development, combined with his strategic insights and practical advice, has been invaluable in overcoming challenges and refining the project's outcomes.

I am also immensely thankful to the volunteers who generously provided their CVs as samples and references for my project. Their willingness to share such personal and professional documents was crucial in training and validating the CV extraction component of InterviewAI. Without their contributions, it would have been significantly more challenging to develop a system capable of accurately processing and analyzing diverse CV formats and content. Their support has directly contributed to the robustness and reliability of the project, and I am truly grateful for their trust and participation.

Equally important, I extend my heartfelt appreciation to the volunteers who dedicated their time to participate in testing InterviewAI, whether as interviewers or interviewees. Their involvement was essential in evaluating the system's emotion detection and question-generation capabilities in real-world scenarios. By providing constructive feedback and engaging with the model during testing phases, they helped identify areas for improvement and ensured the system could deliver context-aware and meaningful interactions. Their commitment and enthusiasm were vital to the iterative development process, and I am deeply appreciative of their contributions.

In conclusion, the success of InterviewAI would not have been possible without the collective support of my supervisor, Dr. Aun Yichiet, and the volunteers who contributed their CVs and time to testing. Their encouragement, expertise, and willingness to collaborate have been the cornerstone of this project's development. I am profoundly grateful for their belief in my vision and their invaluable contributions, which have not only shaped InterviewAI but also inspired me to continue exploring the potential of artificial intelligence in innovative and impactful ways.

ABSTRACT

InterviewAI is an advanced AI-driven platform designed to transform the recruitment process by integrating emotion detection, automated CV analysis, and dynamic question generation powered by large language models (LLMs). The primary goal of this Final Year Project (FYP2) is to enhance recruitment efficiency, fairness, and personalization by extracting critical information from CVs, analyzing candidates' real-time emotional states, and generating tailored interview questions based on these insights. The system employs a sophisticated combination of machine learning models, including a Convolutional Neural Network (CNN) for real-time emotion detection from video feeds and the Llama3 model for context-aware question generation, seamlessly integrated into a unified framework. Leveraging Artificial Intelligence for data processing and Human-Computer Interaction principles for user-centric design, the methodology ensures robust handling of multimodal data, enabling the system to adapt dynamically to each candidate's emotional and professional profile.

Compared to its initial development in FYP1, InterviewAI in FYP2 has been significantly refined to improve accuracy in emotion detection, enhance CV extraction capabilities, and optimize question relevance through iterative model training and user feedback. Final results demonstrate the system's ability to reduce recruitment bias, streamline the interview process, and provide HR professionals with an intelligent tool that adapts to individual candidate responses, thereby fostering a more inclusive and equitable interview experience. By alleviating the administrative burden on recruiters and promoting objective evaluations, InterviewAI showcases substantial potential to revolutionize modern recruitment practices, making them more efficient, unbiased, and tailored to the unique needs of each candidate.

Area of Study: Artificial Intelligence, Machine Learning
Keywords: Emotion Detection, CV Analysis, Question Generation, Artificial Intelligence, Natural Language Processing, Machine Learning, Convolutional Neural Network, Large Language Models, Recruitment Automation, Bias Reduction

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
CV	Curriculum Vitae
HR	Human Resource
JAFFE	Japanese Female Facial Expression
LLM	Large Language Model
NLP	Natural Language Processing
OCR	Optical Character Recognition
PDF	Portable Document Format
RAF-DB	Real-world Affective Faces Database
SVM	Support Vector Machine

Chapter 1 Introduction

This chapter provides a comprehensive overview of the problem statement, motivation, objectives, project scope, and contributions of this research. The extraordinary advancements in artificial intelligence (AI) and machine learning have reshaped numerous industries, introducing unprecedented levels of automation, precision, and adaptability. However, the recruitment sector continues to rely heavily on traditional methods, which are often time-consuming, prone to human biases, and lack personalization. These inefficiencies can lead to suboptimal hiring decisions, negatively impacting both organizations and candidates. To address these challenges, this study introduces InterviewAI, an innovative AI-powered interview assistant designed to revolutionize the hiring process. By integrating emotion-driven question generation, real-time candidate response analysis, and automated CV extraction, InterviewAI aims to enhance the efficiency, fairness, and inclusivity of recruitment. This research seeks to deliver a transformative solution that not only streamlines hiring practices but also fosters a more equitable and candidate-centric experience, aligning with the evolving demands of modern workforce management.

1.1 Problem Statement and Motivation

In the rapidly evolving landscape of modern organizations, the recruitment process is pivotal in building high-performing teams by identifying and securing top talent. However, traditional interview methods present significant obstacles that undermine the effectiveness and fairness of this process. A primary challenge is **the intense anxiety and discomfort experienced by many candidates** during interviews. This emotional strain often impairs their ability to articulate their skills, experiences, and potential, preventing them from presenting their true capabilities. Consequently, organizations risk overlooking highly qualified candidates, resulting in missed opportunities for both the employer, who may fail to hire the best fit for the role, and the candidate, who may be unfairly excluded from career advancement.

Furthermore, **traditional interviews are susceptible to biases**, either conscious or unconscious that exacerbate inequities in the hiring process. These biases can manifest in various forms, such as preconceived notions about a candidate's background, appearance, communication style, or their ability to perform under pressure. Such factors often lead interviewers to make subjective judgments that prioritize superficial traits over relevant qualifications and competencies. This is particularly detrimental for candidates from underrepresented or non-traditional backgrounds, who may face additional barriers in conforming to conventional expectations. As a result, the recruitment process can become perpetuating systemic inequalities and limiting diversity within organizations. The combined impact of interview-related anxiety and entrenched biases creates a flawed system that fails to consistently identify and nurture talent, hindering organizational growth and innovation.

The InterviewAI project is driven by a commitment to address these critical challenges through the development of an innovative, AI-powered solution designed to transform the interview experience. By integrating advanced technologies, InterviewAI aims to **foster a more inclusive, empathetic, and equitable recruitment process**. A key feature of the system is its ability to incorporate emotion-based questioning and real-time emotional analysis. By monitoring a candidate's emotional state, such as detecting signs of stress, nervousness, or sad, the AI can dynamically adjust the interview's pace and question difficulty. For example, if a candidate appears fear, the system might pose simpler, open-ended questions to help them relax and build confidence before progressing to more complex topics. This adaptive approach creates a supportive environment that empowers candidates to perform at their best, alleviating the undue pressure often associated with traditional interviews.

Another motivation of InterviewAI is to **mitigate the impact of biases in the hiring process**. Human interviewers, despite their best intentions, are prone to subjective impressions influenced by factors unrelated to a candidate's suitability for the role. These biases can skew evaluations, leading to inconsistent and unfair outcomes. InterviewAI addresses this by leveraging AI-driven analysis to evaluate candidates based on the substance and quality of their responses, rather than subjective or irrelevant criteria. The system ensures a standardized and objective assessment framework, reducing the influence of factors such as a candidate's appearance, accent, or stress management skills. By focusing on measurable competencies and qualifications, InterviewAI promotes fairness and enhances the likelihood of selecting candidates who are truly best suited for the role.

In short, the InterviewAI project seeks to revolutionize the recruitment landscape by delivering a scalable, efficient, and emotionally intelligent solution that redefines how interviews are conducted. By prioritizing candidate well-being and fairness, InterviewAI not only improves the candidate experience but also enables organizations to make better-informed hiring decisions. This innovative approach has the potential to create a more diverse and inclusive workforce, break cycles of inequality, and drive organizational success by ensuring that talent is recognized and valued for its true potential.

1.2 Objectives

The InterviewAI project is designed to revolutionize the recruitment process by developing an AI-driven system that delivers a personalized, empathetic, and adaptive interview experience. The system integrates advanced technologies, including computer vision and large language models (LLMs), to address the limitations of traditional interviews. The following objectives outline the core components of the project, aimed at enhancing the precision, inclusivity, and effectiveness of the interview process.

1) **Develop an advanced PDF parsing system for comprehensive CV analysis**
   A primary objective is to create a highly accurate and efficient PDF parsing system to extract critical information from candidates' CVs. This system will focus on retrieving essential details, such as personal information, academic qualifications, professional experience, technical and soft skills, and other relevant attributes. By systematically processing and organizing this data, the system will establish a foundation for generating tailored interview questions that align closely with each candidate's unique background. This capability will ensure that the interview process is highly relevant, focused, and capable of assessing candidates based on their specific qualifications, thereby improving the overall quality and precision of recruitment outcomes.
2) **To develop an emotion recognition method to identify the candidates' emotions in real time**
   Another key objective is to develop a state-of-the-art emotion recognition system that leverages computer vision technologies to monitor and analyze

candidates' emotional states in real time. By employing facial recognition techniques and deep learning models, the system will accurately identify a range of emotions, including happiness, sadness, surprise, anger, and neutrality, during the interview. This real-time emotional analysis will enable the system to adapt the interview dynamically, creating a more engaging and supportive experience that responds to the candidate's emotional needs. By fostering an environment where candidates feel understood and at ease, the system will help them perform to their fullest potential, enhancing the fairness and effectiveness of the evaluation process.

3) **Build a Dynamic Question Generation Framework Powered by Pretrained LLMs**
The project aims to design an innovative and flexible question generation framework that harnesses the capabilities of pretrained large language models (LLMs) to create contextually relevant and adaptive interview questions. This framework will integrate data from the candidate's CV and real-time emotional feedback to dynamically tailor the nature, complexity, and technical depth of questions. By aligning questions with the candidate's qualifications and emotional state, the system will deliver a personalized and engaging interview experience that balances challenge with support. This approach ensures that questions are not only pertinent to the candidate's background but also conducive to a positive and productive interaction, leading to a more comprehensive and equitable assessment of their suitability for the role.

1.3 Project Scope and Direction

The scope of this project is centred on the development and implementation of a sophisticated AI-driven interview system that leverages emotion detection and large language models (LLMs) to generate personalized interview questions. This project is divided into several key phases.

1) **Automatic Extraction of Key Information from CVs**
The project involves the process of design an efficient PDF parsing system capable of extracting essential information from candidates' CVs. This system will focus on retrieving critical details, including personal information, academic qualifications, work history, technical skills, and other pertinent

attributes. By accurately processing and structuring this data, the system will provide a solid foundation for generating highly relevant and customized interview questions that align with the candidate's unique background, thereby improving the precision and effectiveness of the interview process.

2) **Create an Emotion Detection System Leveraging RAF-DB Dataset**

This phase is dedicated to integrating emotion detection capabilities into the InterviewAI system to elevate the recruitment process. The project will utilize the Real-world Affective Faces Database (RAF-DB), a comprehensive and well-regarded dataset, to establish a robust framework for recognizing and classifying a diverse range of emotional states, such as happiness, sadness, anger, disgust, neutral, and more. By leveraging the rich and varied emotional annotations provided by RAF-DB, the system will train and optimize a state-of-the-art emotion detection model using advanced computer vision techniques and deep learning algorithms. This model will undergo extensive fine-tuning to achieve high precision in detecting and interpreting candidates' emotional cues in real time during the interview process. The resulting emotional insights will play a critical role in enabling the system to generate interview questions that are both contextually appropriate and sensitive to the candidate's emotional state, creating a more supportive, engaging, and personalized interview experience that empowers candidates to showcase their true potential.

3) **Create an Advanced Emotion Recognition System for Real-Time Analysis**

This phase involves building a cutting-edge emotion recognition system that leverages computer vision techniques to monitor and interpret candidates' emotional states in real time. Utilizing deep learning models and facial recognition technology, the system will identify a range of emotions, such as neutral, anger, happiness, disgust and sadness during the interview. By analyzing these emotional cues, the system will enable a more adaptive and engaging interview experience, ensuring that the questioning process is responsive to the candidate's emotional needs and fosters a supportive environment.

4) **Design a Dynamic Question Generation Framework Powered by Pretrained LLMs**

The project is to develop a question generation framework that harnesses pretrained large language models (LLMs) to produce interview questions tailored to the candidate's profile and emotional state. This framework will dynamically adjust the type and technical depth of questions based on the candidate's CV data and real-time emotional feedback and reply from candidates. By creating a context-aware and flexible questioning strategy, the system aims to deliver a personalized interview experience that is both relevant to the candidate's qualifications and sensitive to their emotional well-being, ensuring an optimal balance of challenge and support.

1.4 Contributions

The InterviewAI project significantly advances the field of AI-driven recruitment by introducing a pioneering system that leverages advanced Large Language Models (LLMs) to create a highly personalized and emotionally responsive interview process. By integrating Llama 3 with real-time emotion detection, the system dynamically generates interview questions that adapt to a candidate's emotional state, such as happiness, sadness, or nervousness. This ensures questions are not only contextually relevant but also tailored to reduce candidate discomfort, addressing a common challenge in traditional interviews. This emotionally intelligent approach fosters a supportive environment, enabling candidates to perform at their best and promoting fairer, more accurate assessments of their potential.

Another key contribution lies in the automation of CV analysis, which streamlines the recruitment process for HR professionals. The system employs an efficient PDF parsing mechanism to extract critical information from candidates' CVs, including personal details, educational qualifications, professional experience, and skills. This automated process informs the generation of contextually appropriate interview questions, ensuring alignment with each candidate's unique background. By reducing the manual effort required for data extraction, the system allows HR teams to manage large candidate pools more effectively, freeing them to focus on strategic tasks such as candidate engagement and decision-making.

The project also enhances candidate evaluations by incorporating non-verbal communication analysis, utilizing datasets like RAF-DB to interpret facial expressions and emotional cues. By combining computer vision techniques with LLM-driven question generation, the system evaluates both verbal responses and non-verbal signals, such as expressions of joy or discomfort. This holistic approach provides a more comprehensive and nuanced understanding of candidates, capturing insights that traditional interviews might miss. By integrating emotional intelligence into the evaluation process, the system improves the accuracy and inclusivity of hiring outcomes, setting a new standard for recruitment practices.

1.5 Report Organization

The InterviewAI project report is structured to provide an overview of the development, implementation, and evaluation of an AI-driven interview assistance system, with each chapter focusing on a distinct aspect of the project. Chapter 1 introduces the project by presenting the problem statement, which highlights the challenges of traditional interviews, such as candidate anxiety and interviewer bias, and outlines the objectives to create an empathetic, fair interview process using AI technologies. It also defines the project scope and contributions, offering a clear picture of the system's goals, including emotion detection, resume parsing, and dynamic question generation, and its potential impact on recruitment practices. Chapter 2 conducts a literature review of existing similar systems, analyzing their strengths and weaknesses, and discusses the hardware and software requirements for InterviewAI, alongside a review of Large Language Models (LLMs), Convolutional Neural Networks (CNNs), and resume parsing techniques to establish the theoretical foundation and technological context for the project.

Chapter 3 focuses on system methodology, employing various diagrams to illustrate how the InterviewAI system is built and operates, including a system architecture diagram to show the overall structure, a use case diagram to depict user interactions, and an activity diagram to detail the workflow of the interview process. Chapter 4 delves into system design, providing a detailed system block diagram, component design, and flowchart to explain the system's functionality and the interconnections between its modules, while guiding users on the specific tools and steps needed to develop the project. Chapter 5 covers the implementation phase, guiding users through hardware and software setup, system settings and configuration, the overview of the system operation and addressing implementation issues and challenges.

Chapter 6 evaluates the system's performance through testing and discussion, analyzing metrics like emotion detection accuracy and question relevance, presenting A/B testing results to compare the system with and without emotion detection, and discussing challenges and objective fulfillment to assess the project's success. Finally, Chapter 7 concludes the report with a summary of the project's achievements, reflecting on how InterviewAI addresses traditional interview challenges and enhances recruitment through AI-driven features, and provides recommendations for future improvements. All of the chapters provide a cohesive narrative of the InterviewAI project, from its conceptualization and design to its implementation, evaluation, and future potential, offering a thorough understanding of its development.

Chapter 2 Literature Review

2.1 Hardware

The hardware involved in this project is the computer. For this AI-powered interview system project, hardware infrastructure is essential to support the complex operations associated with machine learning, particularly when using advanced technologies such as Large Language Models (LLMs) and emotional detection algorithms. The hardware selection planned to ensure optimal performance throughout the development, testing and implementation.

Description	Specifications
Model	HP Pavillon Laptop 14-bf1xx
Processor	Intel(R) Core (TM) i5-8250U
Operating System	Windows 11 Pro
Graphic	NVIDIA GeForce GT 940MX
Memory	16GB DDR4 RAM
Storage	250GB SATA HDD

Table 2.1.1 Specifications of Laptop

2.2 Software

Description	Specifications
Programming Language	Python
Markup Language	Html, Flask
Integrated Development Environment (IDE)	Visual Studio Code (VSCode)
AI model	Llama 3.2 Vision (11B, Instruct, Q8_0)

Table 1.2.1 Software Requirement

2.3 Large Language Models (LLMs)

Large Language Models (LLMs) represent a transformative class of artificial intelligence systems that utilize deep learning architectures to process and generate human-like text. Trained on vast and diverse datasets, LLMs capture intricate patterns, contextual nuances, and linguistic structures, enabling them to excel in a range of natural language processing (NLP) tasks, including text generation, translation, summarization, and question-answering[Brown]. The advent of LLMs has significantly influenced various industries by facilitating automation and addressing complex language-related challenges. Notable examples include OpenAI's GPT-4, which has redefined benchmarks in text coherence and contextual relevance across applications, and Google's BERT-based models, which have advanced contextual understanding in areas such as search optimization and conversational AI. These advancements have broadened the scope of NLP, making previously intractable tasks feasible [4]. Among these models, Llama 3, developed by Meta AI, stands out as a significant milestone in LLM evolution. Building on its predecessors, Llama 3 offers enhanced scalability, diverse training data, and robust performance in multilingual and multimodal tasks, making it particularly suitable for applications like InterviewAI, which leverages Llama 3.2 Vision for real-time, emotion-sensitive question generation. The model's ability to integrate text and vision inputs aligns with the growing trend of multimodal LLMs, positioning it as a key tool for innovative recruitment solutions.

2.3.1 Llama 3

Large Language Models (LLMs) have transformed natural language processing (NLP) by leveraging advanced deep learning architectures to generate, interpret, and contextualize human-like text, driving innovation across domains such as education, healthcare, and recruitment. Llama 3, developed by Meta AI, represents a significant milestone in LLM evolution, building on its predecessors with enhanced scalability, diverse training data, and multimodal functionality. Launched in 2024, Llama 3 comprises models with 8 billion (8B), 70 billion (70B), and 405 billion (405B) parameters, utilizing a dense Transformer architecture optimized for efficiency and performance across tasks like text generation, translation, summarization, and question-answering [16]. The Llama 3.2 release introduced multimodal variants, notably llama3.2-vision:11b-instruct-q8_0, which integrates text and image processing to support applications such as visual question answering and emotion detection [1]. Trained on a massive corpus of approximately 15 trillion multilingual tokens, over eight times the 1.8 trillion tokens used for Llama 2 and Llama 3 achieves exceptional linguistic diversity and contextual understanding, making it a versatile tool for global and multimodal applications [16]. The development of Llama 3 reflects broader trends in LLM research, where increased model size, enriched training datasets, and multimodal integration have expanded NLP's scope. Unlike earlier models limited to monolingual or text-only processing, Llama 3's multilingual and vision-enabled variants address the need for AI systems that operate across diverse linguistic and sensory contexts [5]. For instance, its vision capabilities enable InterviewAI to detect candidate emotions in real-time, tailoring questions to foster a supportive interview environment. This aligns with research emphasizing that multimodal LLMs enhance recruitment by integrating verbal and non-verbal cues, such as facial expressions, to improve evaluation accuracy and candidate comfort [6]. Llama 3's open-source availability under the LLaMA 3 Community License democratizes access to advanced AI, enabling students and researchers to innovate without the financial barriers of proprietary models [1].

2.3.2 Strengths and Weaknesses

Llama 3 offers numerous strengths that position it as a leading model for **advanced NLP and multimodal applications**, particularly for InterviewAI, which utilizes llama3.2-vision:11b-instruct-q8_0 for real-time recruitment tasks. Its robust multilingual capabilities enable processing and generation of text in languages such as English, German, French, Italian, Spanish, and others Team. This proficiency is vital for global recruitment platforms like InterviewAI, where candidates from diverse linguistic backgrounds require tailored, language-appropriate questions. By supporting seamless communication across multiple languages, Llama 3 eliminates the need for separate language-specific models, reducing complexity and costs for international organizations Team. This enhances InterviewAI's inclusivity and ensures equitable candidate experiences regardless of native language, a key factor in modern recruitment systems. Multilingual LLMs bridge linguistic barriers, expanding access to AI-driven applications in diverse global contexts [5].

The open-source nature of Llama 3, under the LLaMA 3 Community License, provides a significant advantage over proprietary models like GPT-4 or Claude, particularly for resource-constrained users such as students and researchers [15]. Unlike closed-source models requiring costly licenses, Llama 3 **is freely accessible for non-commercial use**, enabling customization without financial barriers [1]. For InterviewAI, this facilitated deployment of llama3.2-vision:11b-instruct-q8_0 on a school server using the Ollama framework, allowing a student-led project to leverage advanced AI without licensing costs [15]. Open-source models democratize AI access, empowering educational institutions and small-scale projects to innovate in NLP and recruitment [1]. Open-source multimodal LLMs reduce dependency on proprietary platforms, enabling tailored solutions at lower costs [6]. This fosters collaborative development, as seen in InterviewAI's Ollama integration, and supports fine-tuning for tasks like question generation [15].

Despite these strengths, Llama 3 has notable weaknesses, primarily its **substantial computational resource requirements**, which challenge users with limited hardware. With parameter counts from 8 billion to 405 billion, and llama3.2-vision:11b-instruct-q8_0 requiring 10-15GB of GPU memory, Llama 3 demands high-performance hardware like NVIDIA A100 or RTX 3090 GPUs for efficient training and inference Stubell. For students using standard laptops without accelerators, running Llama 3 locally is impractical, limiting experimentation and deployment Stubell. In InterviewAI, this is mitigated by a school server with GPU support, but dependency on such infrastructure highlights and accessibility barrier for educational settings. The computational intensity of multimodal LLMs restricts adoption in resource-constrained environments, often requiring cloud-based solutions [6]. Large-scale LLMs also raise sustainability concerns due to high energy and resource demands [5].

To address these challenges, **pre-trained Llama 3 models**, specifically the llama3.2-vision:11b-instruct-q8_0 variant, are deployed via the Ollama framework, leveraging optimized weights to eliminate the need for computationally expensive training from scratch [15]. By hosting these models on a centralized school server equipped with robust GPU capabilities, systems can efficiently process large volumes of text data, such as documents, user queries, or generated responses, without requiring high-end local hardware, thereby democratizing access to advanced AI in resource-constrained environments like educational institutions [15]. The school server's GPU resources provide the critical computational power to meet Llama 3's intensive inference demands, enabling seamless execution of tasks like text structuring, question generation, and content analysis with high efficiency [6]. The use of pre-trained models, combined with quantization techniques like Q8_0, significantly enhances computational efficiency by reducing the model's memory footprint and accelerating inference times, allowing robust performance on shared server infrastructure Stubell. Quantization, such as the Q8_0 configuration, optimizes the model's size and processing speed, making it feasible to deploy large-scale LLMs on standard server hardware without compromising accuracy or responsiveness 10. This server-based approach also facilitates collaborative access, enabling multiple users to leverage the same GPU resources simultaneously, which is particularly advantageous in academic settings or small-scale organizations with limited budgets [11]. Moreover, centralized deployment mitigates the need for individual users to maintain costly local infrastructure, streamlining system maintenance and updates [13]. Further advancements in model optimization, such as enhanced quantization, pruning, and knowledge distillation, hold promise for reducing computational demands even further, improving scalability for resource-limited environments and enabling broader adoption of AI-driven systems [2].

2.4 Emotion Detection

Emotion detection, a rapidly evolving field within artificial intelligence, focuses on identifying human emotions through physiological signals, primarily facial expressions. This technology harnesses deep learning models to analyze visual data, detecting patterns associated with emotional states such as anger, disgust, fear, happiness, sadness, surprise, and neutrality. Its significance is particularly evident in human resources (HR), where understanding emotional cues during interviews or interactions provides valuable insights into individuals' sentiments and responses [2]. Recent advancements have led to systems that utilize facial landmarks—such as eyes, eyebrows, and mouth, and to classify emotions with high precision and efficiency, making them essential for applications requiring rapid and reliable emotion assessment [2]. Various algorithms are employed to categorize data linked to emotional states. Traditional machine learning methods, including Support Vector Machines (SVMs), Decision Trees, and Naive Bayes classifiers, have been used to process manually extracted features from facial images. However, deep learning techniques, notably Convolutional Neural Networks (CNNs), have transformed emotion detection by autonomously capturing complex facial patterns [14].

These advancements have significantly improved the robustness of emotion detection systems, enabling accurate identification of subtle emotional cues across diverse populations and environmental conditions. Comprehensive datasets have been critical in addressing challenges related to data diversity and model generalization [9]. In HR, these systems enhance candidate evaluation by interpreting emotional responses, fostering more empathetic and informed interactions [14]. However, the computational complexity of deep learning models requires substantial hardware resources, posing challenges for deployment in resource-constrained settings [7]. Ongoing research explores optimization techniques to enhance accessibility and efficiency, broadening the applicability of emotion detection systems [7].

2.4.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) have revolutionized emotion detection by automatically extracting relevant features from raw facial image data through multiple convolutional layers. These layers capture hierarchical patterns, progressing from simple edges and textures in early layers to complex structures like facial landmarks and muscle movements in deeper layers, enabling precise representation of emotional expressions [8]. From research, Mehmet A. and his team outline a CNN-based pipeline for emotion detection. The process begins with a camera-captured input image, followed by face detection using the Haar Cascade classifier, which identifies facial regions by recognizing patterns resembling human facial features. The detected regions are cropped to isolate facial features, converted to grayscale to minimize computational complexity, and resized to a uniform 64x64 pixel dimension for consistent neural network processing [8].

The preprocessed images undergo convolutional and max pooling layers. Convolutional layers apply operations to extract features, while max pooling layers reduce spatial dimensions, retaining only the most prominent features to enhance efficiency. The resulting feature maps are flattened and fed into fully connected layers, which integrate learned features to make decisions about the input image. A SoftMax classifier predicts the probability distribution across seven emotion classes which are Happy, Sad, Neutral, Angry, Disgust, Surprised, and Afraid, selecting the class with the highest probability as the predicted emotion [8]. This approach leverages deep learning's ability to process complex visual data, improving accuracy in diverse scenarios [9]. Advanced CNN architectures, incorporating attention mechanisms to prioritize critical facial regions, have further enhanced performance, particularly in challenging real-world conditions [14].

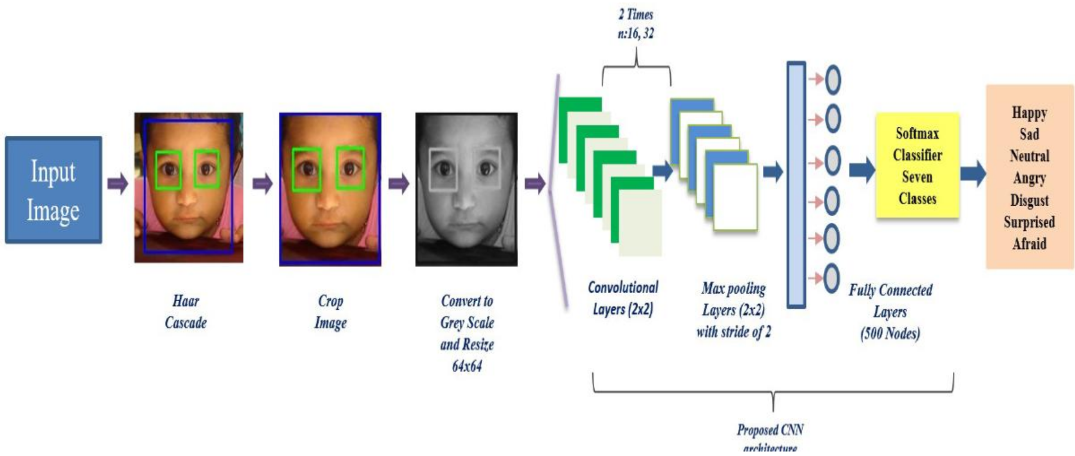


Figure 2.4.1Process of CNN

2.4.2 Strength and Weakness of CNN

CNN-based models have **demonstrated high accuracy on datasets** like JAFFE, RAF-DB showcasing their robustness in emotion detection tasks. These models outperform traditional machine learning methods, such as SVMs, which depend on manual feature extraction and struggle with the nuances of facial expressions [8]. CNNs utilize multiple convolutional and pooling layers to distinguish emotions like anger, disgust, happiness, sadness, and surprise, automatically learning hierarchical features from low-level edges to complex facial contours. This autonomous learning enables CNNs to capture subtle variations in expressions, essential for accurate emotion classification [8]. By learning features directly from training data, CNNs eliminate the need for handcrafted features, focusing on intricate patterns critical for emotion detection [9].

In HR applications, CNNs enhance candidate evaluation by accurately detecting emotional responses, supporting informed hiring decisions. Datasets like FER, FERPlus, RAF-DB and AffectNet contain thousands of labeled facial images across various emotions, demographics, and lighting conditions, provide a robust foundation for training, improving model fairness and accuracy [9]. The integration of transfer learning in CNNs further boosts performance by leveraging pre-trained models, reducing training time and data requirements [14]. These capabilities make CNNs suitable for diverse applications, such as psychological assessments and customer service, where nuanced emotion detection is critical [14].

However, **CNNs are susceptible to overfitting, particularly when trained on small or biased datasets, limiting their generalization to new data**. In emotion detection, an overfitted model may excel on its training dataset but fail to classify emotions accurately in unseen facial images, especially if the data lacks diversity in demographics, expressions, or environmental conditions [8]. This limitation is significant in HR, where diverse candidate profiles require robust performance [14]. Additionally, CNNs **demand large volumes of labeled data** to perform effectively, particularly for emotion detection, where subtle facial changes are critical. Acquiring such datasets is costly and time-consuming, as manual labeling adds significant complexity [9]. The computational intensity of CNNs also poses deployment challenges, requiring high-performance hardware like GPUs, which may be inaccessible in low-resource settings [7]. Real-time emotion detection in dynamic environments, such as live interviews, exacerbates these computational demands [7].

To mitigate overfitting, **techniques like data augmentation and regularization are employed**. Data augmentation expands the training dataset through transformations such as rotation, flipping, or scaling, enabling CNNs to learn generalized representations of facial expressions. Regularization methods, including dropout and weight decay, impose constraints during training to reduce model complexity, preventing the learning of irrelevant details [8]. These strategies enhance model robustness, particularly for HR applications requiring diverse data [14]. Transfer learning also addresses overfitting by fine-tuning pre-trained CNNs on smaller datasets, improving generalization with limited data [14].

To address the need for large labeled datasets, **pre-labeled datasets provide comprehensive, annotated facial images** covering emotions such as happiness, sadness, anger, and surprise. These datasets reduce manual labeling efforts, improving the efficiency, accuracy, and fairness of CNN-based emotion detection systems [9]. Emerging datasets, such as RAF-DB, further enhancing model robustness across real-world conditions [9].

2.5 Text Extraction

The automation of text extraction from CVs is a vital component of modern recruitment systems, addressing the inefficiencies of handling large volumes of job applications. Manual CV review is labor-intensive, time-consuming, and prone to human errors, which can introduce biases and compromise hiring decisions. Automated text extraction technologies enable rapid identification and retrieval of key information, such as applicants' names, contact details, educational qualifications, work experience, skills, certifications, and other relevant details. In the context of InterviewAI, PDF parsing is employed as the primary method to process digitally created CVs, ensuring accurate data extraction to support emotion-driven question generation and candidate evaluation. This approach streamlines recruitment workflows, reduces administrative burdens, and promotes a more equitable and efficient screening process by leveraging structured data for personalized interview experiences [10].

2.5.1 PDF Parsing

PDF parsing is a specialized technology designed to extract text and structured data from Portable Document Format (PDF) files, which are widely used in recruitment due to their consistent formatting across diverse platforms and devices. PDFs encapsulate a complex internal structure comprising text, images, tables, metadata, and other elements, which parsing tools interpret to retrieve embedded content [3]. Tools such as PyMuPDF, PyPDF2, and PDFMiner are engineered to navigate this structure, accessing document metadata, text objects, fonts, and layout instructions to extract text efficiently. These parsers are particularly effective for machine-generated PDFs, where text is stored as character codes rather than graphical elements, enabling rapid and direct extraction. For example, a digitally created CV with well-defined sections such as personal information section, education section, work experience section can be processed by traversing the document's object hierarchy, ensuring precise extraction of structured data for integration into InterviewAI's question-generation pipeline 10. This capability is critical for processing standard resumes and other formal documents that adhere to predictable formats, facilitating seamless data integration into automated recruitment systems.

PDF parsing tools operate by first reading the document's metadata to understand its structure and then interpreting its objects, which include text, fonts, and layout instructions. This process allows parsers to differentiate between text, images, and other elements like tables or graphs, ensuring accurate extraction of textual content. Advanced parsers can also handle embedded annotations, hyperlinks, and formatting details, enriching the extracted data for downstream applications. In recruitment, where CVs often follow standardized templates, PDF parsing enables the extraction of key fields, such as job titles, educational degrees, or skill sets that are essential for tailoring interview questions to candidates' profiles. Moreover, these tools support batch processing, allowing InterviewAI to handle multiple CVs simultaneously, which is crucial for high-volume recruitment scenarios.

2.5.2 Strengths and Weaknesses of PDF Parsing

PDF parsing offers significant advantages in terms of **speed, accuracy, and efficiency when processing digitally generated PDFs**. By directly accessing embedded text, parsers eliminate the need for resource-intensive conversion steps, enabling rapid processing of large document volumes which is a critical requirement in high-throughput recruitment scenarios [11]. For instance, PyMuPDF can extract text from a standard CV in seconds, making it an ideal tool for handling the majority of CVs submitted in PDF format. Additionally, advanced parsers preserve the logical structure of the document, including section headings, bullet points, and hierarchical organization, which enhances the accuracy of downstream data processing and ensures that extracted information aligns with the CV's intended structure. This structural fidelity is particularly beneficial for InterviewAI, as it enables the system to map extracted data to specific candidate attributes, facilitating context-aware question generation [Brown].

However, PDF parsing **faces challenges with non-standard PDFs**, such as those with complex layouts or varying encoding standards. Documents created using design platforms like Canva often feature multi-column formats, embedded tables, or stylized fonts, which can disrupt the logical flow of text extraction, leading to jumbled or incomplete outputs. Additionally, differences in PDF creation tools and encoding standards may cause inconsistencies, such as misread characters or formatting errors, reducing the reliability of extracted data 10. These issues can result in partial extractions, such as incomplete job titles or misaligned section data, which may compromise the accuracy of automated processing.

To address the limitations of PDF parsing, a robust **post-processing strategy can be employed**, utilizing document classification, regex-based cleaning, and pattern-based validation to ensure reliable text extraction across diverse document formats. Initially, documents are classified based on their structure by analyzing metadata and initial text samples, which helps identify complex layouts or encoding issues early in the process. For documents with multi-column formats or stylized fonts, the extracted text can be processed by advanced server-side models capable of interpreting varied layouts, ensuring accurate structuring of data despite parsing challenges. Regular expression (regex) techniques are applied to clean extracted outputs, correcting common formatting errors such as trailing commas or inconsistent quote usage, thereby addressing inconsistencies caused by varying encoding standards.

Furthermore, a validation layer is implemented to ensure the completeness and accuracy of extracted data. This involves cross-referencing extracted fields against predefined patterns, such as expected formats for names, dates, or section headers, to identify and correct anomalies. For instance, if a job title is partially extracted due to a complex layout, the validation process can flag the error and apply fallback rules, such as extracting key fields from the raw text using simple pattern matching. This approach minimizes errors without requiring manual intervention. By combining document classification, regex-based cleaning, server-side processing, and pattern-based validation, this strategy overcomes the inherent limitations of PDF parsing and ensures reliable text extraction across a wide range of document types encountered in real-world scenarios.

2.6 Existing System – Paradox's Olivia

Artificial intelligence (AI) has revolutionized recruitment through systems like Paradox's Olivia, an AI interview chatbot designed to automate key hiring tasks such as resume screening, candidate pre-screening, and interview scheduling. Utilizing pre-trained large language models (LLMs) on cloud servers, Olivia engages candidates via text-based interactions, extracting qualifications from resumes and responding to queries without incorporating emotion detection, prioritizing efficiency in sectors like retail, hospitality, and healthcare. Its cloud-based architecture supports the computational demands of processing large candidate volumes, making it a prominent example of AI-driven recruitment technology. Olivia's focus on functional automation aligns with the broader landscape of interview chatbots that streamline administrative tasks while maintaining accessibility for diverse organizational needs. This review evaluates Olivia's capabilities, analyzing its strengths, limitations, and potential improvements to understand its role and opportunities for enhancement in recruitment and related applications.

Olivia's strengths significantly enhance recruitment processes by **improving efficiency and candidate engagement**. It reduces resume screening time by up to 70%, allowing recruiters to process high-volume applications in under 24 hours, as evidenced by industry reports. The cloud-hosted infrastructure enables scalability, supporting thousands of simultaneous interactions, which is essential for large-scale hiring in fast-paced industries [12]. These strengths highlight Olivia's ability to optimize hiring workflows, offering speed, scalability, and accessibility that make it a leading solution in AI interview chatbot technology.

Despite its strengths, Paradox's Olivia faces significant limitations that undermine its performance in recruitment applications, including the **absence of real-time emotion detection, difficulties processing image-based or non-standard resumes, generic LLM outputs, and reliance on cloud infrastructure**. The lack of real-time emotion detection restricts Olivia's ability to interpret and respond to candidate emotional states, such as anxiety or confidence, which are critical for assessing soft skills like emotional intelligence and adaptability in roles requiring interpersonal interaction, such as customer service or healthcare positions. This limitation reduces engagement, as candidates may feel the interaction is mechanical, and hampers the depth of assessments, as recruiters miss nuanced insights into candidate demeanor, a concern highlighted in studies on AI-driven recruitment. Additionally, Olivia struggles with image-based or non-standard resumes, such as scanned PDFs or creatively formatted CVs, which compromises screening reliability. This issue leads to incomplete or erroneous extraction of qualifications and experience, potentially overlooking qualified candidates or requiring manual intervention, which negates the automation's efficiency in high-volume hiring scenarios. Furthermore, Olivia's generic LLM outputs lack industry-specific nuance, producing broad responses that fail to address the unique requirements of specialized roles, such as technical questions for software engineers or clinical scenarios for nurses. This reduces the relevance of candidate interactions and screening questions, diminishing the system's effectiveness in identifying role-specific competencies, particularly in diverse sectors like technology or healthcare. Finally, Olivia's dependence on cloud infrastructure escalates operational costs, as continuous cloud usage incurs significant expenses, and introduces latency during peak usage periods, such as large-scale recruitment drives, which disrupts real-time interactions and candidate experience. These delays can frustrate candidates and recruiters, especially in fast-paced industries like retail, where timely responses are critical, as noted in scalability studies.

To address the limitations of Paradox's Olivia, including its lack of real-time emotion detection, challenges with unstructured data, generic LLM outputs, and cloud dependency, several targeted solutions can be implemented to enhance its performance in recruitment applications. **Integrating a Convolutional Neural Network (CNN)**, trained on datasets like the Real-world Affective Faces Database (RAF-DB), can enable real-time emotion detection, allowing Olivia to adapt responses based on candidate emotional states, such as anxiety, thereby improving engagement and assessment depth, as supported by studies on AI-driven recruitment. To improve resume processing, **advanced PDF parsing** techniques, such as those using libraries like pdfplumber or PyMuPDF, can accurately extract data from both text-based and image-based resumes. **Fine-tuning the LLM** on industry-specific datasets can enhance response relevance for specialized roles, ensuring questions and interactions reflect sector-specific differences. **Transitioning from cloud to a local server-based infrastructure**, equipped with GPU capabilities, reduces operational costs and latency during peak usage, as the project scale does not yet require cloud-level resources, load balancing and caching strategies further optimize server performance. CNN-based emotion detection, advanced PDF parsing, LLM fine-tuning, and server-based deployment are collectively address Olivia's weaknesses, enhancing its adaptability, accuracy, and efficiency for diverse recruitment scenarios without relying on costly cloud infrastructure.

Technology/Method	Weakness	Solution
Llama 3	substantial computational resource requirements	Utilizing pre-trained models from providers like Ollama
Emotion Detection using CNN	Prone to overfitting	Data augmentation and regularization methods
	Requires large amounts of labelled data	Using pre-labelled datasets
Text Extraction using PDF parser	Faces challenges with non-standard PDFs	Post-processing strategy can be employed

Table 2.6.1 Weakness and Solution of Technologies

Existing System	Weakness	Solution
Paradox's Olivia	Absence of real-time emotion detection generic LLM outputs	Integrating a Convolutional Neural Network (CNN)
	Difficulties processing image-based or non-standard resumes	Advanced PDF parsing
	Generic LLM outputs	Fine-tuning the LLM
	Reliance on cloud infrastructure.	Transitioning from cloud to a local server-based infrastructure

Table 2.6.2 Weakness and Solution of Existing System

Chapter 3 System Methodology/Approach

3.1 Methodology

In the realm of human resources, particularly within the recruitment process, a system capable of dynamically engaging with candidates and delivering meaningful feedback is highly valuable. The proposed project seeks to create an advanced system designed to support HR professionals by generating customized interview questions using real-time emotion detection and data derived from the candidate's resume. By integrating technologies such as facial emotion recognition, natural language processing (NLP) and machine learning, this system aims to streamline portions of the interview process, enhancing its efficiency and personalization for each candidate. The core functionality of the system is to produce interview questions that are tailored to the candidate's skills and experience while also adapting to their emotional state during the interaction. This responsive and adaptive method is intended to improve the overall quality of engagement, making interviews more interactive, insightful, and beneficial for both HR professionals and candidates.

Data Collection Phase

This phase centers on gathering the data needed to train the models that drive the system.

For emotion detection, a dataset of labeled images showcasing various human emotions is essential. The Real-world Affective Faces Database (RAF-DB), which is publicly available, fits the bill perfectly. It contains thousands of facial images captured in real-life settings, labeled with emotions like happiness, sadness, anger, fear, surprise, disgust, and neutral. These images reflect natural scenarios and offering a solid base to train a model that can accurately spot emotions during live interviews. This variety ensures the model can handle different lighting, angles, and backgrounds, which is key for real-world applications.

For the CV extraction model, a diverse set of CVs is required to cover a range of formats. This includes everything from digital PDFs to scanned documents and image-based resumes. Such diversity helps the model learn to process all kinds of documents, whether they're neatly typed or a bit messy, ensuring accurate extraction of details. The dataset trains the model to identify sections like personal info, education, work experience, and skills, preparing it to handle the variety of CVs candidates might submit.

Model Training Phase

Once the data is collected, the training phase kicks in to prepare the models that enable the system's core features. This step is crucial for ensuring the system can accurately detect emotions, extract CV details, and generate tailored interview questions. The question-generation capability relies on the Llama 3 model from Ollama, which uses the advanced Transformer architecture. This setup, with its encoder-decoder structure, automatically analyzes CV data and detects emotions to create interview questions. The encoder examines the data, capturing connections through self-attention, while the decoder crafts questions that match the candidate's profile. Features like multi-head attention and positional encoding ensure the questions are contextually relevant and personalized, such as asking a software developer about their experience with a specific coding framework. This approach excels at processing large amounts of data, making the questions feel spot-on for each candidate.
A Convolutional Neural Network (CNN) model is also trained using the RAF-DB dataset to enable real-time emotion detection. CNNs are excellent for this task because they can spot patterns in images, like the way facial features shift with different emotions. Training involves multiple rounds, feeding the model batches of facial images with their emotion labels, ensuring it can handle real-world variations like dim lighting or different facial angles.
Additionally, the resume analysis model uses techniques like PDF parsing to extract key details from CVs efficiently. Llama 3 then steps in to analyze this data, identifying specifics like the candidate's name, education, university, and skills, providing a clear summary for interviewers to use before the conversation begins.

Implementation Phase

The implementation phase involves combining the trained emotion detection model, CV extraction model, and question generation model into a single system ready for real world use.
The emotion detection model integrates into the system to analyze facial expressions in real-time via a webcam during interviews. A video processing pipeline continuously

captures video frames and feeds them into the model, allowing it to track and update the candidate's emotional state as the interview progresses. If a candidate appears nervous, the system can adjust its approach, perhaps by asking a simpler question to ease their tension.
Meanwhile, the CV extraction model handles automated processing of CVs, extracting details like personal info, education, work experience, and skills. This information flows to the question generation model, fine-tuned with Llama 3, to create questions that fit the candidate's background.
The question generation model then combines the CV data with real-time emotion input to produce personalized interview questions that align with the candidate's profile and emotional state, ensuring a more engaging and responsive interview experience.

Testing Phase

The testing phase focuses on verifying the system's reliability, accuracy, and overall performance, a vital step before real-world deployment.
CV information extraction testing involves running the system with various CVs to check how well it extracts details like names, education, work experience, and skills across different formats. Results are compared to manually extracted data to gauge accuracy, ensuring the model can handle diverse document types effectively.
Emotion detection testing uses videos of candidates displaying different emotions to evaluate the model's real-time performance. The RAF-DB dataset's real-world variety helps test the model under different conditions, comparing its outputs to expected emotions to confirm accuracy and reliability.
Question generation testing examines the system's ability to create relevant, personalized questions based on CV data and detected emotions. Scenarios with different CVs and emotional states are simulated to review the questions, ensuring they match the candidate's background and emotional context, like asking a calmer question if stress is detected.
Overall, end-to-end testing ensures all components work together smoothly, with performance monitored to identify any issues or bottlenecks that need addressing.

3.2 System Design Diagram

3.2.1 System Architecture Diagram

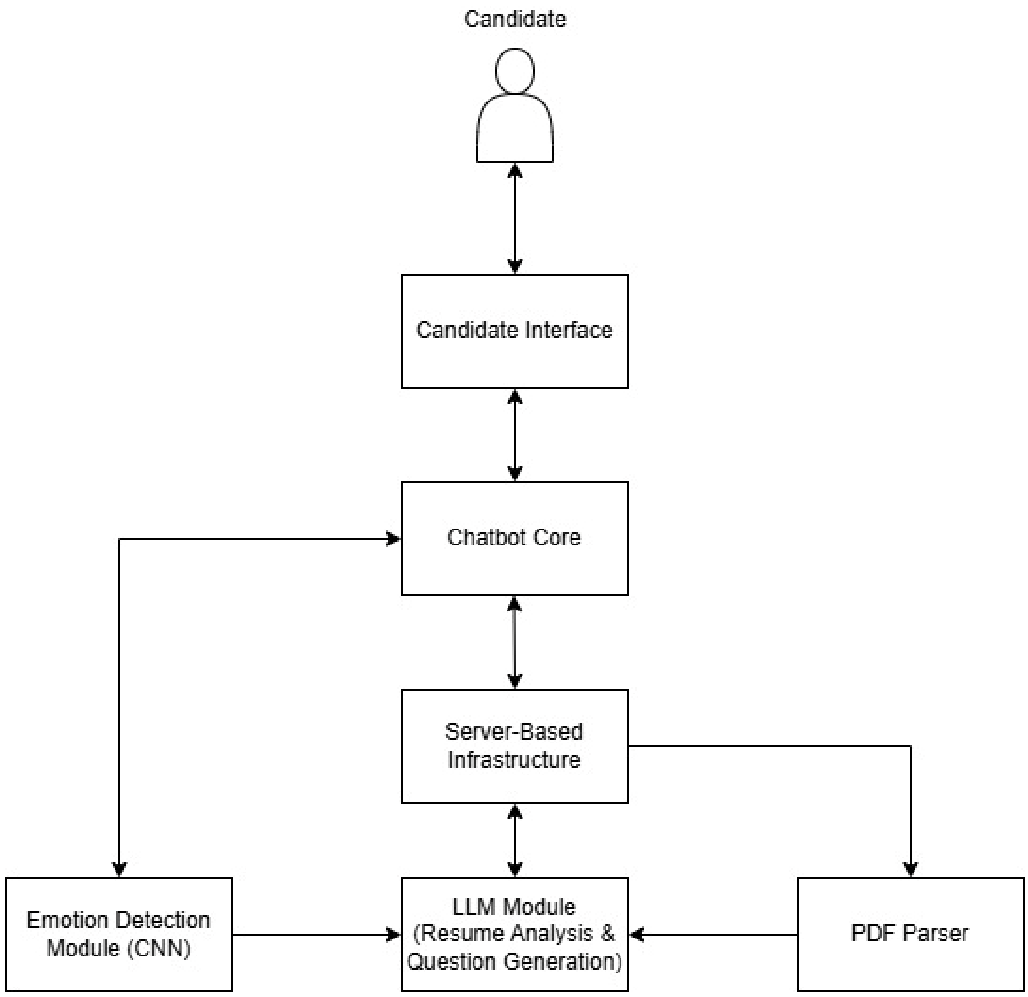


Figure 3.2.1 System Architecture Diagram

The System Architecture Diagram is organized in a layered and hierarchical manner, reflecting the flow of data from the user interaction layer to the core processing modules and back. At the highest level, the **candidate** represents the primary user interacting with the system. The candidate engages with the system by providing essential inputs, such as a resume and video feed, and receiving outputs in the form of tailored interview questions and prompts. This interaction occurs through the **Candidate Interface**, a user-facing front-end module designed to facilitate seamless communication between the candidate and the system's backend components. The Candidate Interface is responsible for capturing inputs, such as textual responses, uploaded resume files, and live video streams from a webcam, and delivering outputs, including dynamically generated interview questions, to the candidate in an intuitive and accessible manner.

Directly beneath the Candidate Interface lies the **Chatbot Core**, which serves as the central coordinator of the system. The Chatbot Core acts as an intermediary, managing the flow of data between the Candidate Interface and the various processing modules. It ensures that inputs from the candidate are appropriately routed to the relevant modules for processing and that the resulting outputs, such as generated questions, are sent back to the Candidate Interface for presentation. The Chatbot Core plays a pivotal role in maintaining the system's responsiveness, orchestrating the interactions between components to deliver a cohesive and adaptive interview experience.

All processing modules are encapsulated within the **Server-Based Infrastructure**, a critical component that provides the computational resources necessary for the system's real-time operation. The Server-Based Infrastructure hosts the core modules responsible for data processing, analysis, and question generation, ensuring that the system can handle the computational demands of advanced algorithms such as Convolutional Neural Networks (CNNs) and large language models (LLMs). This infrastructure is designed to support high-performance processing, leveraging server resources such as CPUs and GPUs to enable rapid data analysis and question generation, which is essential for maintaining a natural conversational flow during interviews.

Within the Server-Based Infrastructure, three primary processing modules are depicted:

- **PDF Parser**: This module is tasked with processing the Candidate's resume to extract structured data, such as personal details, educational qualifications, work experience, and skills. The extracted data is then forwarded to the LLM Module for further analysis, forming the foundation for personalized question generation.

- **Emotion Detection Module (CNN)**: This module utilizes a Convolutional Neural Network (CNN) trained on the Real-world Affective Faces Database (RAF-DB) to detect the Candidate's emotional state in real-time.. It receives video input from the Candidate Interface via a webcam and captures a photo only when answering questions, emotion such as neutral, happiness, fear, sadness, or surprise is detected, optimizing resource usage while enabling adaptive question generation. The detected emotions are then sent to the LLM

Module, enabling the system to adapt its questioning strategy based on the Candidate's emotional responses, thereby enhancing the interview's effectiveness and sensitivity to the Candidate's state.

- **LLM Module (Resume Analysis & Question Generation)**: This module leverages the Llama 3 model, a state-of-the-art large language model, to perform two key functions: resume analysis and question generation. The module receives structured data from the PDF Parser and emotion labels from the Emotion Detection Module. It analyzes the resume data to identify key attributes, such as the candidate's skills, experience, and educational background, and combines this information with the detected emotions to generate personalized interview questions. The questions are designed to be relevant to the candidate's profile while also being sensitive to their emotional state to ensure more engaging and supportive interview experience.

Data Flow and Interactions

The interactions between components are represented by bidirectional arrows, which indicate the flow of data throughout the system. The candidate provides inputs through the Candidate Interface, which include the resume file, video feed, and textual responses to questions. These inputs are relayed to the Chatbot Core, which distributes them to the appropriate processing modules within the Server-Based Infrastructure for analysis and processing.

The resume data is sent to the **PDF Parser**, which extracts structured information and forwards it to the **LLM Module** for analysis. Concurrently, the video feed is transmitted to the **Emotion Detection Module (CNN)**, which processes the video frames to detect the Candidate's emotional state and sends the resulting emotion labels to the LLM Module. The LLM Module integrates the extracted resume data and emotion labels to generate tailored interview questions, which are then sent back to the Chatbot Core. The Chatbot Core delivers these questions to the Candidate Interface, where they are presented to the Candidate, completing the data flow cycle.

This bidirectional flow ensures that the system can continuously process inputs and adapt its outputs in real-time, a critical requirement for maintaining a natural and responsive interview process. The Chatbot Core plays a crucial role in this cycle, ensuring that data is seamlessly integrated and that the system operates as a cohesive unit. The absence of a database aligns with the project's design, as the system relies on temporary storage within the server environment to manage data during the interview process, reducing latency and ensuring real-time performance.

3.2.2 Use Case Diagram

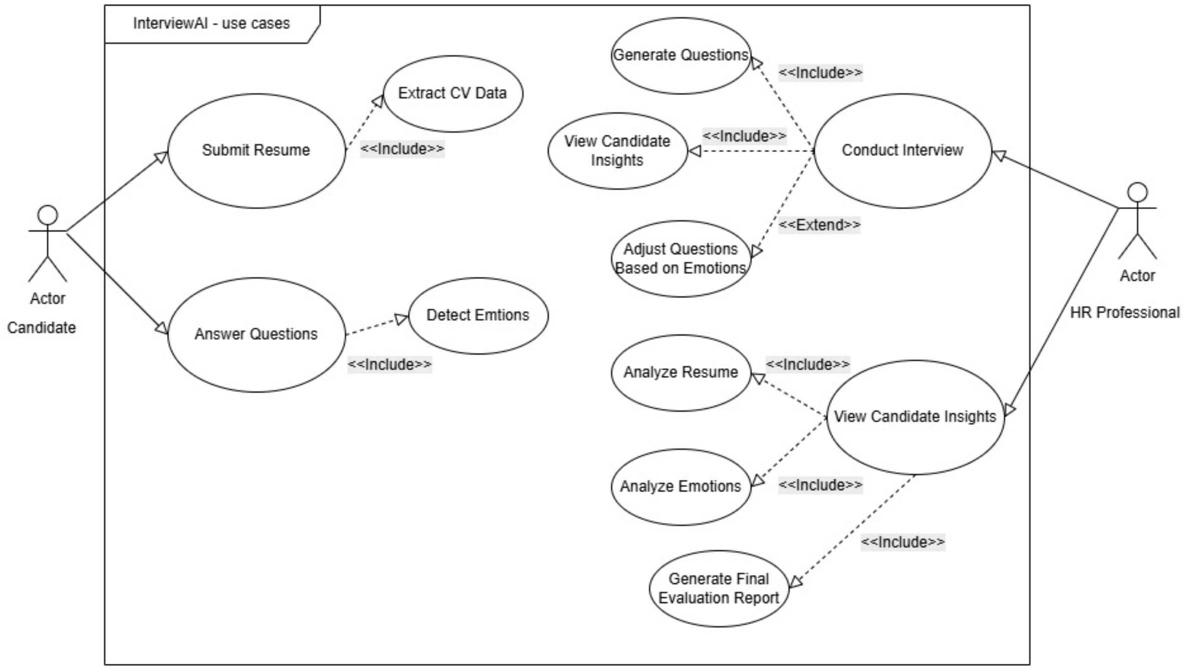


Figure 3.2.3 Use Case Diagram

The Use Case Diagram for the InterviewAI system serves as a critical tool in elucidating the system's functionalities and the interactions between its primary actors—the Candidate and the HR Professional and the system itself, which provides a structured representation of the processes involved in conducting an automated and adaptive interview for recruitment purposes. This diagram encapsulates the system's core objectives, which include automating the interview process, generating personalized questions based on the Candidate's professional background and emotional state, and delivering actionable insights to support hiring decisions.

The diagram identifies two primary actors: the Candidate, who engages with the system to participate in the interview, and the HR Professional, who leverages the system to evaluate the Candidate's suitability for a role. The Candidate interacts with the system through the Submit Resume use case, which initiates the interview process by allowing the Candidate to upload their resume, a critical step in enabling the system to gather foundational data about the Candidate's qualifications. This use case includes the Extract CV Data subprocess, where the system processes the uploaded resume to extract structured information such as personal details, educational background, work experience, and skills, utilizing the PDF Parser module within the Server-Based Infrastructure. The Extract CV Data use case further includes Analyze Resume, a subprocess that interprets the extracted data to identify key attributes, which are essential for generating relevant interview questions. This hierarchical structure ensures that the system comprehensively understands the Candidate's profile before proceeding with the interview, aligning with the project's goal of personalizing the interview experience based on the Candidate's background.

Another significant interaction for the Candidate is the Answer Questions use case, which enables the Candidate to respond to the system's dynamically generated questions, forming the core of the interview process. This use case includes the Detect Emotions subprocess, where the Emotion Detection Module (CNN) monitors the Candidate's emotional state using video input from a webcam. The Detect Emotions use case has been optimized to capture a photo when answering the questions, reducing computational load while maintaining the system's ability to adapt questions based on emotional cues. The Detect Emotions use case further includes Analyze Emotions, where the system interprets the detected emotions (e.g., neutral, happiness, fear, sadness, surprise) using a Convolutional Neural Network (CNN) trained on the Real-world Affective Faces Database (RAF-DB). This analysis allows the system to understand the Candidate's emotional responses, providing valuable context for question adaptation and ensuring a more empathetic and engaging interview experience.

The Conduct Interview use case is a central functionality that both the Candidate and the HR Professional interact with, encapsulating the entire interview process from question generation to response collection. This use case includes the Generate Questions subprocess, where the LLM Module (Resume Analysis & Question Generation) leverages the analyzed resume data to produce tailored interview questions, ensuring relevance to the Candidate's professional background. The Generate Questions use case also includes Analyze Resume, reinforcing the dependency on resume data for question generation. Additionally, the Adjust Questions Based on Emotions use case extends Generate Questions, indicating an optional behavior where the system modifies questions dynamically based on emotion detected. This adaptive capability is a cornerstone of the InterviewAI system, enabling it to respond to the Candidate's emotional state in real-time and enhance the interview's effectiveness, as outlined in the project's objectives.

Both the Candidate and the HR Professional can engage with the View Candidate Insights use case, which provides a summarized view of the Candidate's performance, including their responses and emotional states throughout the interview. The HR Professional, however, has a more direct and functional interaction with this use case, using it to evaluate the Candidate's suitability for the role by reviewing detailed insights such as response quality, emotional consistency, and overall fit. In addition, this use case includes the generation of a Final Evaluation Report, which consolidates all assessment metrics into a structured summary, offering both qualitative and quantitative data to aid in the final decision-making process. The View Candidate Insights use case also extends Conduct Interview, signifying that these insights are a result of the interview process itself. This linkage provides a comprehensive and seamless assessment experience for the HR Professional, underscoring the system's dual role in conducting interviews and delivering meaningful, actionable outcomes.

The Use Case Diagram effectively captures the InterviewAI system's modular design, illustrating how its components work together to achieve a seamless and adaptive interview process.

3.2.3 Activity Diagram

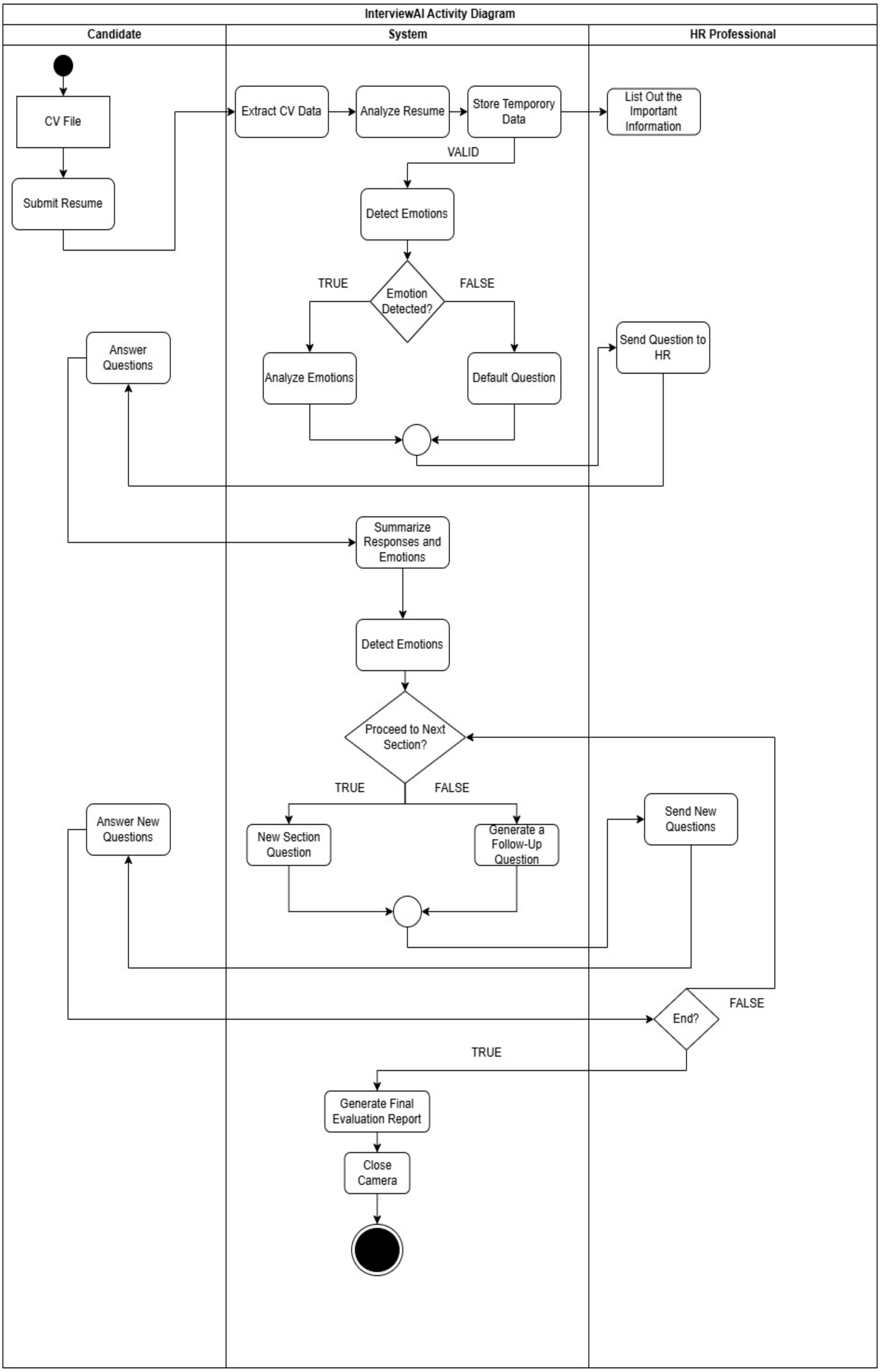


Figure 3.2.3 Activity Diagram

The InterviewAI system workflow starts with the Candidate submitting their resume. The Candidate uploads their CV file through the system's interface, and this action hands the process over to the System, which begins by extracting important details from the resume, such as the Candidate's education, work experience, skills, and any other relevant information that can help shape the interview. After extracting this data, the System analyzes the resume to better understand the Candidate's background, looking at things like their job history or technical skills to make sure the questions asked later are relevant to their experience. Since the system doesn't use a database, it stores this information temporarily in memory, ensuring everything happens in real-time without saving anything permanently, which is how the project was designed. At the same time, the System prepares a summary of the key details from the resume and sends it to the HR Professional, giving them an early look at the Candidate's profile so they can follow along with the interview process and understand who they're evaluating.

Once the resume is processed, the System moves on to checking the Candidate's emotions using a webcam, a step that helps make the interview more personalized. The System uses its Emotion Detection Module to watch the Candidate and only takes a photo when it notices a change in their emotions, like if they go from looking happy to seeming nervous, which is an improvement we made to save on resources. If the System detects an emotion, happiness, sadness or fear, it analyzes what that emotion means to get a better sense of how the Candidate is feeling. If it doesn't detect any emotion, it skips this analysis and instead sends a default question to the HR Professional for a quick review before asking the Candidate. After this, the System sends the first interview question to the Candidate, who then responds with their answer. The Candidate's response is sent back to the System, which keeps track of both the answer and any emotions detected during this interaction, making sure it has all the information it needs to keep the interview moving forward in a way that feels natural and tailored to the Candidate.

With the Candidate's answer, the System summarizes both the response and any emotions it detected, putting together a clear summary of how the Candidate is doing so far. This summary helps the System decide what to do next, whether that's adjusting the question type or preparing for a new section. The System then checks for any emotional changes by looking at the Candidate again through the webcam. If it notices a change in emotion, like if the Candidate now seems more confident, the System will adjust the question type based on this updated emotional state—for example, asking a more open-ended question to encourage the Candidate to share more, ensuring the interview stays relevant and supportive. If there is no emotional change, the System will instead generate a follow-up question based on the Candidate's previous answer, asking something that digs deeper into what they said, like requesting more details about a project they mentioned. The interview is structured into four sections—Initial, Personal, Life, and Technical. The HR Professional can decide when to move to the next section by clicking 'next', giving them control over the interview's progression. These new or follow-up questions are sent back to the Candidate, who answers them, and the System keeps this cycle going, constantly adapting the interview to the Candidate's responses and emotions to make the experience as meaningful as possible.

When the interview is done, the System generate a final evaluation report for HR Professional as a reference. After that, the system closes the webcam since it no longer needs to watch the Candidate's emotions, saving resources and ending that part of the process

Chapter 4 System Design

4.1 System Block Diagram

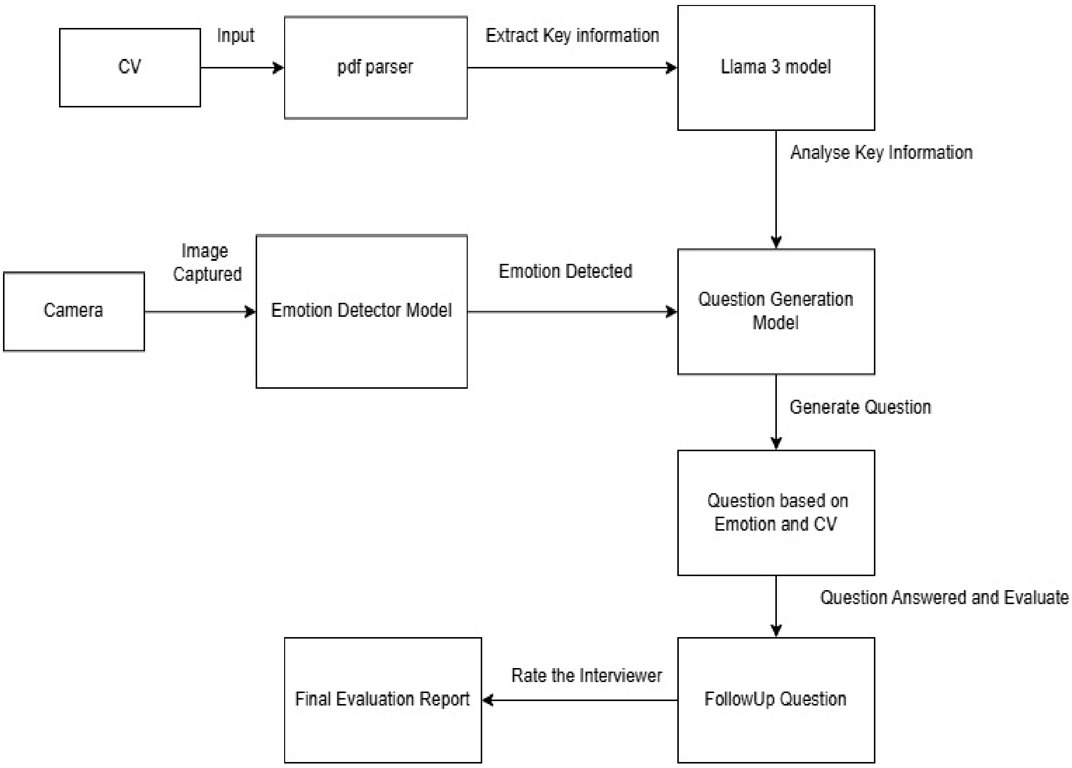


Figure 4.1.1 System Block Diagram of the Project

4.2 System Components Specification

The InterviewAI system consists of several key components that work together to deliver an automated, adaptive, and personalized interview process for recruitment, ensuring that questions are tailored to the Candidate’s professional background and emotional state. These components operate in real-time without a database, relying on temporary storage within the Server-Based Infrastructure, and support a structured interview format with four sections—Initial, Personal, Life, and Technical, where transitions between sections are controlled by the HR Professional via a 'next' button. Each component is designed to handle a specific task, from processing the Candidate’s resume to generating and evaluating questions, ensuring the system meets the project’s objectives of enhancing recruitment efficiency through intelligent automation.

The first component, the PDF Parser, is responsible for extracting structured data from the Candidate’s resume to enable personalized question generation. It takes a CV file in PDF format, uploaded through the Candidate Interface, and outputs structured data such as personal details (e.g., name, contact information), education (e.g., degrees, institutions), work experience (e.g., job titles, companies, duration), and skills (e.g., technical proficiencies). The PDF Parser uses Python-based libraries like PyMuPDF or pdfplumberto process files up to 5 MB.

Following the PDF Parser, the Llama 3 Model analyzes the extracted CV data to understand the Candidate's background for question generation. It receives the structured CV data, including personal details, education, work experience, and skills, and uses this information along with the detected emotions to generate questions. This component leverages the Llama 3 model, a large language model optimized for natural language understanding, running on a local server with GPU acceleration.

The Emotion Detector Model monitors the Candidate's emotional state to enable adaptive question generation, capturing a photo only when a significant emotional change is detected, as optimized for resource efficiency. It takes video frames from the Candidate Interface via a webcam and outputs emotion labels (e.g., happiness, sadness, fear, surprise, neutral) along with a captured photo when an emotional shift occurs, which is then sent to the Question Generation Model. This component uses a Convolutional Neural Network (CNN) trained on the Real-world Affective Faces Database (RAF-DB), implemented with TensorFlow or PyTorch, and relies on OpenCV for video frame processing. The Emotion Detector Model must detect emotional changes with at least 80% accuracy and capture a photo within 0.5 seconds of detecting a change, ensuring minimal delay during the interview.

The Question Generation Model generates personalized interview questions by integrating the analyzed CV data and detected emotions, adapting the question type based on the Candidate's emotional state across the four interview sections: Initial, Personal, Life, and Technical. It receives analyzed CV insights (e.g., skills, experience) from the Llama 3 Model and emotion labels with captured photos from the Emotion Detector Model, producing tailored questions with adjusted types to suit the Candidate's emotional state, which are then displayed via the Candidate Interface. This component also uses the Llama 3 model, fine-tuned on a dataset of interview questions and emotional contexts, supporting the four-section structure with transitions controlled by the HR Professional’s 'next' button, ensuring a natural conversational flow.

The Follow-Up Question Module evaluates the Candidate’s answers and generates follow-up questions to explore their responses further to ensure a thorough assessment while adapting to emotional changes. It takes the Candidate’s answers from the Candidate Interface, analyzed CV insights, and updated emotion labels as inputs, producing follow-up questions that align with the Candidate’s previous responses and emotional state, which are sent back to the Candidate Interface. This component utilizes the Llama 3 model for natural language understanding and generation, integrated with the Question Generation Model for consistency.

Lastly, the Final Evaluation Report is generated through a comprehensive analysis of the candidate's resume, emotional responses detected during the interview, and the quality of their replies. For each question, a score is assigned on a scale of 1 to 5, contributing to several key evaluation metrics. These include Overall Emotional Stability, which is based on the consistency and appropriateness of the candidate’s emotional expressions throughout the interview. Overall Fit to Job Requirements is determined by comparing the information extracted from the resume with the provided job scope and the content of the candidate’s responses. Reply Quality is rated based on the clarity, relevance, and completeness of each answer. Confidence Level is assessed by analyzing both the emotional cues and the tone of the replies, identifying whether the candidate presents themselves with confidence. All these metrics are then compiled into an Overall Rating, offering a clear and concise reference to help determine whether the candidate is a good fit for the company. This final report supports informed hiring decisions by providing structured, data-driven insights.

4.3 Component Design

4.3.1 PDF Parser

The PDF Parser component of the InterviewAI system, processes the Candidate’s resume to extract key information for generating personalized interview questions. It starts by reading the PDF file and pulling out all the text from its pages using the PyPDF2 library, combining everything into a single string.

Then, it connects to a server after authenticating with a session token to ensure the communication is secure. It sends the extracted text to the server, asking the Llama 3.2 Vision model to turn it into structured data in JSON format, with categories like name, education, universities, skills, projects, work experience, awards, and languages, making sure each category is a list, even if empty. The server takes up to 40 seconds to respond, and once it does, the system looks for the first proper JSON block in the response, fixing common issues like extra commas or wrong quote marks to make it valid. It keeps only the name and categories that are lists, turning the JSON into a usable format, and logs any errors if the parsing doesn't work, falling back to the basic name and university data it found earlier.

If the data is successfully extracted, it sends this structured information back to the server with instructions to format it into a readable list of strings for display, organized into sections like Name, Universities, Education, Skills, Frameworks, Platforms, Projects, Co-curricular Experiences, Work Experience, Awards, and Languages and noting if any section is empty. The server returns this formatted list, which the system checks to ensure it's a proper list of strings and combines it with the raw data to return both. If the server fails at any point, it logs the issue and returns the basic name and university data as a fallback. This process ensures the PDF Parser reliably extracts and organizes the Candidate's resume details, setting the stage for tailored interview questions.

4.3.2 Emotion Detection Model

The Emotion Detection component in the InterviewAI system processes the Candidate's emotional state using a Convolutional Neural Network (CNN) trained on the RAF-DB dataset, which includes images labeled with emotions such as surprise, fear, disgust, happy, sad, angry, and neutral.

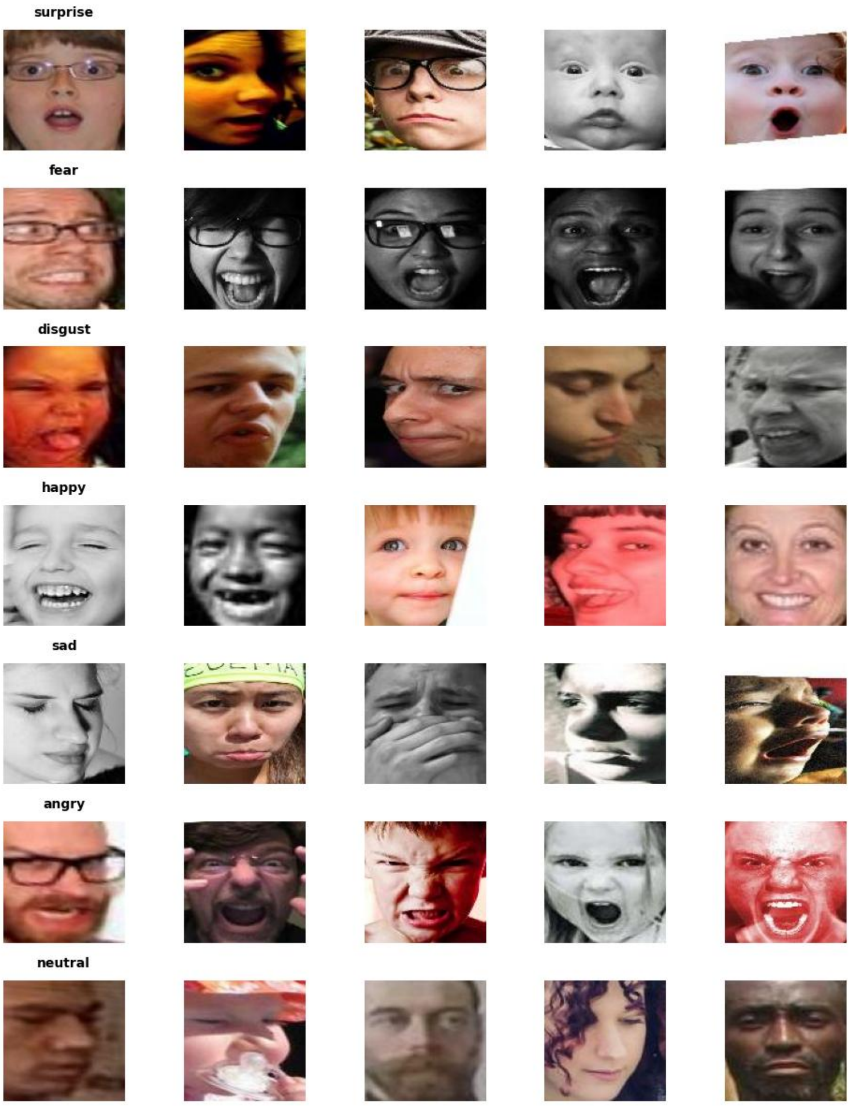


Figure 4.3.1 RAF-DB Dataset

The data preparation starts by loading the training and testing datasets from CSV files using the pandas library, merging them into a single dataset to increase the training data volume. The emotion labels are mapped to numerical values (1 to 7) for processing, and images are loaded from their folders using OpenCV (cv2), converting them from BGR to RGB color format. To balance the dataset, the 'happy' class is reduced to 3,500 images using NumPy for random sampling, as it initially has more samples, and the other classes, surprise, fear, disgust, sad, angry, and neutral are augmented to 3,500 images each using Keras' ImageDataGenerator, applying transformations like rotation, flipping, zooming, and channel shifts to generate diverse samples.

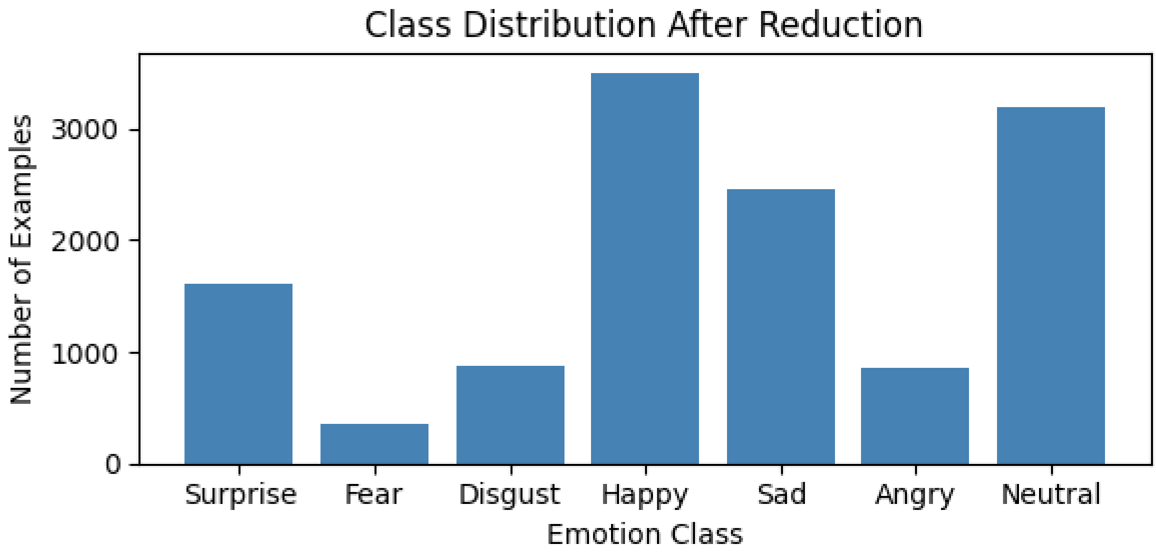


Figure 4.3.2 After 'Happy' Class Reduction

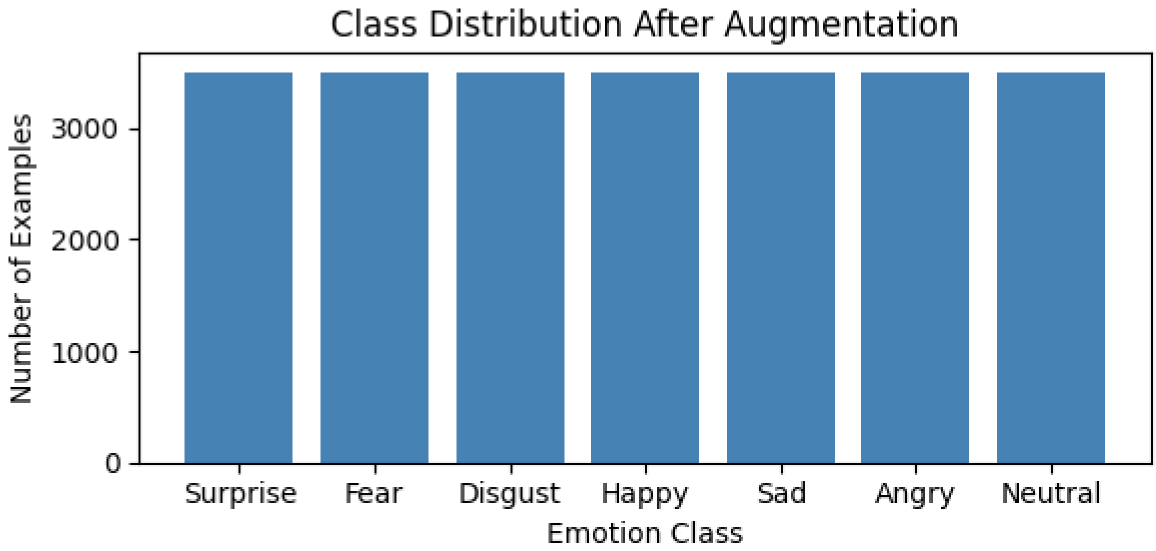


Figure 4.3.3 Class Distribution after Augmentation

The dataset is then split into 75% training and 25% testing sets using scikit-learn's train_test_split function to ensure proper evaluation. Images are normalized by scaling pixel values to a range of 0 to 1 with NumPy and reshaped to a uniform 100x100 pixel size with 3 color channels to fit the CNN's input requirements. During training, real-time augmentation is applied with ImageDataGenerator, including rotation up to 20 degrees, width and height shifts of 10%, and horizontal flips, to improve the model's robustness to variations in facial expressions. Libraries such as Matplotlib and Counter from collections are used to visualize the class distribution before and after balancing, confirming an even spread across emotion classes.

The CNN architecture, built using Keras with TensorFlow as the backend, is designed to effectively capture facial features for emotion detection. It consists of four convolutional layers:

- The first layer has 32 filters of size 3x3 with ReLU activation to detect basic features like edges, followed by a 2x2 max-pooling layer to reduce spatial dimensions and focus on dominant features
- The second layer has 64 filters of 3x3 with ReLU and another 2x2 max-pooling layer to extract more complex patterns
- The third layer increases to 128 filters of 3x3 with ReLU and max-pooling to identify higher-level features
- The fourth layer has 512 filters of 3x3 with ReLU and max-pooling to capture detailed facial characteristics critical for distinguishing emotions.

After these layers, a flatten layer converts the 2D feature maps into a 1D vector, followed by a dense layer with 512 units and ReLU activation to learn complex relationships between features. A dropout layer with a 50% rate is added to prevent overfitting by randomly disabling half of the neurons during training, and the final dense layer has 7 units with a softmax activation to output probabilities for each emotion class.

The CNN is compiled with the Adam optimizer, which adapts the learning rate for faster convergence, and uses categorical cross-entropy loss, appropriate for multi-class classification, while tracking accuracy as the primary metric. Training runs for 60 epochs with a batch size of 64, using the augmented training data generator and the test set for validation.

Fine-tuning is supported by callbacks: ReduceLROnPlateau reduces the learning rate by a factor of 0.1 if validation accuracy stalls for 10 epochs, EarlyStopping stops training if no improvement occurs in validation accuracy for 10 epochs and restores the best weights, and ModelCheckPoint saves the best model based on validation accuracy. The model's depth, with four convolutional layers and increasing filter sizes, ensures it can learn hierarchical features from basic edges to complex facial expressions, while max-pooling and dropout layers help manage computational complexity and generalization, enabling accurate emotion detection for real-time interview adaptation.

4.3.3 Question Generation Model

The Question Generation Model in the InterviewAI system employs a structured technique to create tailored interview questions by leveraging the Candidate's CV data, emotional state, and interview context across four sections: Initial, Personal, Life, and Technical. The process begins by establishing a connection to the with the correct endpoint '/ollama/v1/chat/completions', utilizing the requests library for HTTP communication and the auth module to secure the session with an authentication token, ensuring protected data exchange.

Once connected, the system prepares the CV data by extracting and validating key details such as the Candidate's name, education (e.g., university, degree), technical skills (categorized into skills, frameworks, and platforms), and work experience, ensuring proper formatting even if data is incomplete defaulting to placeholders like "an unspecified university" when necessary. The emotional state, detected previously, is incorporated alongside job details (e.g., job name, scope) and the Candidate's last question and answer to maintain conversational continuity. A prompt is then constructed for the Llama 3.2 Vision model, specifying its role as an HR manager tasked with generating questions that make the Candidate feel comfortable, particularly if emotions like nervousness are detected, by providing supportive acknowledgment before posing a question. The prompt is tailored to the interview section:

- For the Initial section, it requests an introductory question about the Candidate's background.
- For the Personal section, it focuses on strengths, teamwork, or career goals.
- For the Life section, it targets hobbies, stress management, or work-life balance.

- For the Technical section, it generates questions about skills, projects, or problem-solving scenarios relevant to the job scope, always linking to the Candidate's CV, emotion, and prior responses.

The prompt is sent to the server via a POST request with a JSON payload, specifying the model name and a message structure that includes system instructions and user input, with a 10-second timeout to handle potential delays. The system implements a retry mechanism, attempting the request up to two times if it fails due to authentication issues or server errors, logging each attempt for debugging. The server's response is expected to contain four parts: an introductory statement, an emotional acknowledgment, the generated question, and a detailed explanation of why the question was chosen and how it supports the Candidate's emotional state and interview experience, separated by double newlines. The response is parsed, ensuring all parts are present, and if incomplete or erroneous, the system falls back to a default question like "Can you please tell me about yourself?" with an explanation of the failure, logging the issue for further analysis. The Technical section's questions, for instance, might ask about a specific project's challenges if the Candidate's CV highlights relevant experience, adjusting the question's complexity based on detected emotions to ensure the interview remains engaging and supportive.

4.3.4 Follow-Up Question Generation Model

The Follow-Up Question Module in the InterviewAI system generates follow-up interview questions by building on the Candidate's latest answer, emotional state, and overall interview context, ensuring a conversational and adaptive interview experience. The process starts by validating and formatting the Candidate's CV data, including their name, skills which organized into categories like skills, frameworks, platforms, and work experience, using placeholders like "an unspecified company" for missing details to ensure consistency. The system also gathers the job details (e.g., job name, scope), the Candidate's detected emotion (e.g., fear, happy), and previous interactions to provide context for the follow-up question, formatting these into a structured summary for inclusion in the prompt.

A detailed prompt is then constructed for the Llama 3.2 Vision model, specifying its role as an HR manager focused on making the Candidate feel comfortable, especially if emotions like nervousness are detected, by offering supportive, psychology-backed reassurance before asking a question. The prompt includes the Candidate’s name, emotion, job information, previous interactions, and latest answer, instructing the model to generate a response in four parts: an introductory statement referencing the latest answer conversationally (e.g., "Thank you for sharing that, [Name]!"), a brief acknowledgment of the Candidate’s emotion with comforting words, for example, "It’s okay to feel nervous—let’s take it one step at a time", when the system detecting the candidate is nervous, a follow-up question tailored to the latest answer, CV skills, and work experience while aligning with the job role, and an explanation detailing why the question was chosen and how it supports the candidate’s emotional state and interview experience.

The prompt is sent to the server at the endpoint ‘/ollama/v1/chat/completions’ via a POST request using the requests library, with a JSON payload specifying the model and a message structure that includes system instructions and user input, setting a 10-second timeout to handle potential delays. The system implements a retry mechanism, attempting the request up to two times if it fails due to authentication issues or server errors, ensuring robust communication. The server’s response is parsed to extract the four parts, validating their presence and format, and if the response is incomplete or fails, a default follow-up question like “Can you tell me more about your experience or skills relevant to this role?” is used, accompanied by an explanation of the failure, ensuring the interview continues smoothly while maintaining a focus on the Candidate’s emotional state and job-relevant skills.

4.4 Flowchart

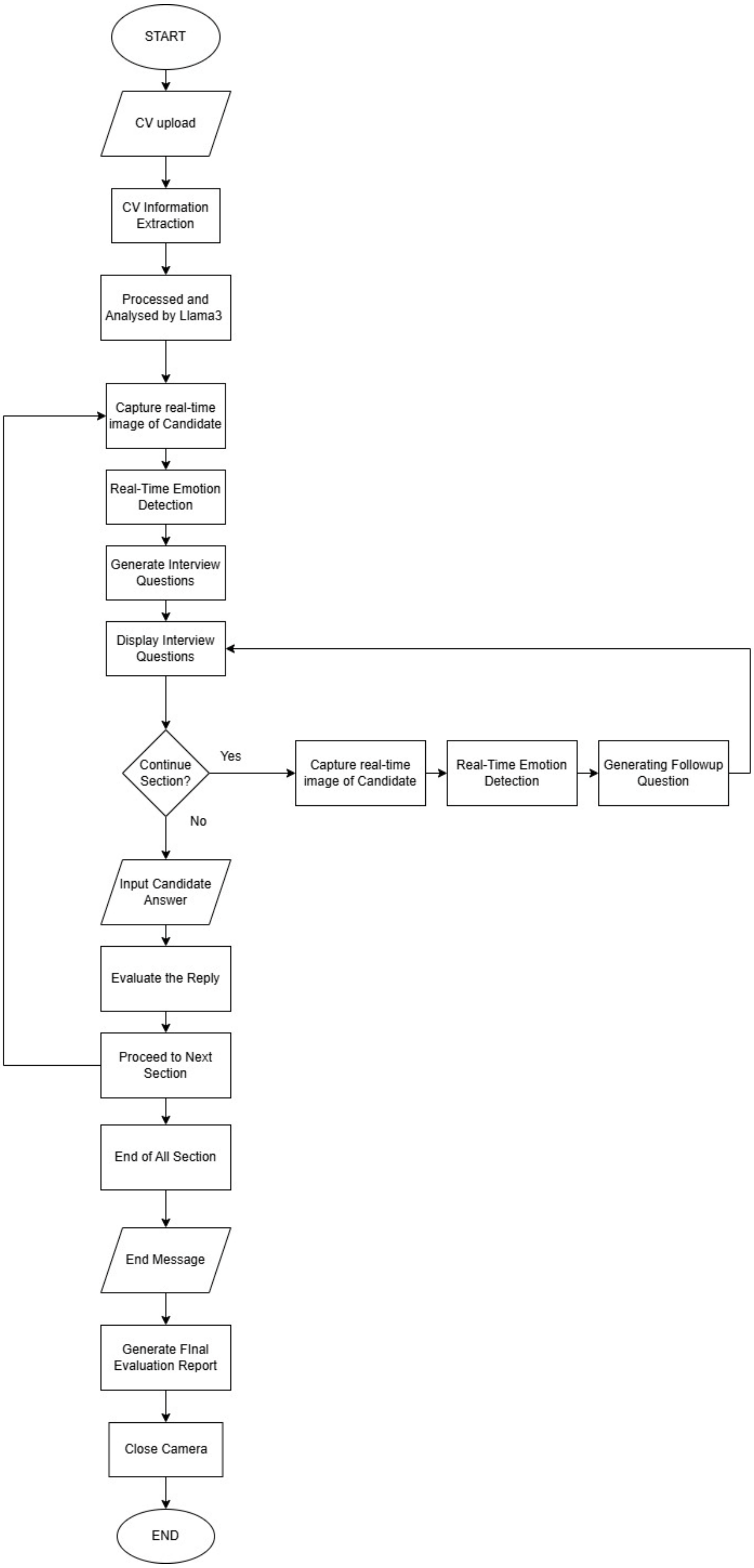


Figure 4.4.1 Image of Flowchart

4.4.1 Explanation of Flowchart

The InterviewAI system initiates its workflow when a Candidate uploads their CV, marking the beginning of a structured and adaptive interview process designed to enhance recruitment efficiency. The system employs PDF parsing techniques to extract details from the CV, then analyzing the document's structure to retrieve text accurately, regardless of diverse layouts or formatting inconsistencies commonly found across resumes. This extracted text, encompassing the candidate's personal and professional information, is then processed and forwarded to the Llama 3 model for in-depth analysis. The Llama 3 model systematically identifies and categorizes critical data points, including the candidate's full name, educational background , such as degrees earned and institutions attended, professional skills, work experience, projects, co-curricular activities, awards, and languages, and ensures a comprehensive understanding of the candidate's qualifications for question generation.

Following the CV analysis, the system activates a webcam to capture a real-time image of the candidate, which is then processed by the Emotion Detection Model to assess their emotional state. This model, a Convolutional Neural Network trained on the RAF-DB dataset, identifies emotions such as happiness, sadness, fear, or neutrality by analyzing facial features, capturing a photo only when a significant emotional change is detected to optimize resource usage The detected emotion, combined with the categorized CV data, is utilized by the Question Generation Model to craft personalized interview questions tailored to the candidate's background and emotional state. These questions are designed to align with the candidate's professional experience and current mood. Accompanying each question, the system provides the interviewer with contextual guidelines and explanations, detailing the rationale behind the question's formulation and how it supports the candidate's emotional well-being and interview experience, thereby facilitating a more informed and empathetic interaction.

The system then presents the HR Professional with the generated question and offers the option to either continue within the current section or advance to the next one, with the interview structured into four distinct sections: Initial, Personal, Life, and Technical, as controlled by the HR Professional via a 'next' button If the decision is to remain in the current section, the Candidate inputs their answer, which the system analyzes alongside their updated emotional state—obtained through another real-time image capture and emotion detection cycle. The Follow-Up Question Module then generates a new question, building on the candidate's response and emotional cues, ensuring a deeper exploration of their skills, experiences, or personal attributes relevant to the section, such as technical problem-solving in the Technical section or stress management in the Life section. Alternatively, if the choice is to proceed to the next section, the system captures a fresh image to reassess the candidate's emotional state, generates a new question appropriate to the upcoming section, and continues the process, maintaining a dynamic and responsive interview flow that adapts to the candidate's evolving emotions and responses.

This iterative process persists through all four sections and each stage of the interview is tailored to the candidate's profile and emotional state. The system facilitates a seamless transition between sections, with the HR Professional controlling progression via the 'next' button and continuously adapts questions to maintain relevance and supportiveness throughout the interview. Upon completion of all sections, the system concludes by displaying an End Message to the candidate. The system also generates a final evaluation report for HR Professional that including the score of the candidates. After that, the system will close the camera automatically, signaling the successful conclusion of the interview process, having provided a comprehensive, emotion-aware, and personalized assessment that aligns with the project's objective of optimizing recruitment through advanced automation and emotional intelligence.

Chapter 5 System Implementation

5.1 Hardware Setup

The InterviewAI system demands a high-performance hardware configuration to support its real-time processing needs, particularly for CV extraction, emotion detection, and question generation. The core system requires a server equipped with a multi-core CPU to handle parallel tasks such as PDF parsing and model inference. It is essential for accelerating the Llama 3 model's natural language processing tasks and the Emotion Detection Model's convolutional neural network (CNN) operations, which include four convolutional layers (32, 64, 128, and 512 filters) for real-time facial emotion analysis. The server should have a minimum of 32 GB of RAM to manage large datasets, such as the RAF-DB dataset used for emotion detection training, and to support temporary storage of CV data, as the system operates without a database.

The InterviewAI system utilizes a single interface for the HR Professional, designed to manage the interview process efficiently. This interface requires a standard computer equipped with a webcam supporting at least 720p resolution at 30 FPS to capture real-time images of the Candidate for emotion detection to ensure accurate facial feature analysis. The computer must have a minimum of 8 GB of RAM and a dual-core CPU to run the interface smoothly, along with a stable internet connection for communication with the server at the '/ollama/v1/chat/completions' endpoint. The HR Professional uses this interface, accessed via a modern web browser to view generated questions, control transitions between the four interview sections (Initial, Personal, Life, Technical) and observe Candidate insights.

5.2 Software Setup

The InterviewAI system depends on a comprehensive software stack to perform its core operations, including CV parsing, emotion detection, and question generation, ensuring a seamless and adaptive interview process. Python 3.8 or higher serves as the primary programming language, managing all system components such as PDF parsing, server communication, and model inference. Key Python libraries include PyPDF2 for extracting text from CVs, pdfplumber as an alternative for handling complex PDF layouts. The requests library enables HTTP communication with the university server at the '/ollama/v1/chat/completions' endpoint.

The Llama 3 model, responsible for CV analysis and question generation, requires TensorFlow 2.10 or PyTorch 1.13 to operate the Llama 3.2 Vision model (11b-instruct-q8_0). The Emotion Detection Model, a CNN trained on the RAF-DB dataset, also leverages TensorFlow or PyTorch, using OpenCV (cv2) for image processing, NumPy for numerical operations, and Keras' ImageDataGenerator for data augmentation during training. Additional libraries for data preparation include pandas for loading CSV files, scikit-learn for splitting datasets, and Matplotlib for visualizing class distributions. The university server hosting the Llama 3 model employs a REST API framework to manage the '/ollama/v1/chat/completions' endpoint, ensuring secure communication through the auth module for token-based authentication.

Development of the InterviewAI system is conducted using Visual Studio Code (VSCode) as the primary integrated development environment, facilitating efficient coding, debugging, and testing of the Python-based components. The HR Professional interface is web-based, developed using Flask as the lightweight Python framework to handle server-side logic, alongside HTML for structuring the front-end interface, ensuring a user-friendly experience. This interface requires a modern browser with JavaScript enabled to support real-time interaction features, such as webcam access for emotion detection and displaying questions across the four interview sections. The combination of these software tools and frameworks ensures the system's compatibility with the university server, enabling a robust and interactive interview experience.

5.3 Setting and Configuration

Setting up and configuring the InterviewAI system requires a series of methodical steps to ensure all components operate seamlessly, supporting its goal of delivering an adaptive and emotion-aware interview experience. The process begins by installing Python 3.8 or higher on the system, followed by the necessary libraries via pip, including PyPDF2 and pdfplumber for PDF parsing, requests for server communication, TensorFlow for model inference, opencv-python for image processing, pandas for data handling, scikit-learn for dataset splitting, and Matplotlib for visualization, ensuring all dependencies are met for CV extraction, emotion detection, and question generation. The RAF-DB dataset, essential for training the Emotion

Detection Model, must be downloaded from Kaggle and stored in a designated directory, providing the image data needed to train the CNN model for emotion recognition. The system must be configured to connect to the university server's endpoint at '/ollama/v1/chat/completions', verifying that the URL is accessible and capable of receiving prompt inputs for the Llama 3.2 Vision model, while ensuring the local computer can link to the server for real-time communication, using the requests library and token-based authentication via the auth module. The CNN model, designed with four convolutional layers (32, 64, 128, 512 filters) and trained on the RAF-DB dataset, must be validated to detect human faces and emotions with high accuracy. The HR Professional interface, which facilitates interactive front-end functionality, is developed using Flask as the Python framework for server-side logic, HTML for structuring the interface, and JavaScript for enabling dynamic features like real-time webcam access and question display across the four interview sections (Initial, Personal, Life, Technical), ensuring a user-friendly and responsive experience.

5.4 System Operation

This section provides a comprehensive and detailed walkthrough of how the Interview AI system operates in practice, breaking down each component and process into clear, step-by-step stages. Through the use of visual aids and annotated screenshots, users will be able to gain a clearer understanding of the inner workings of the system.

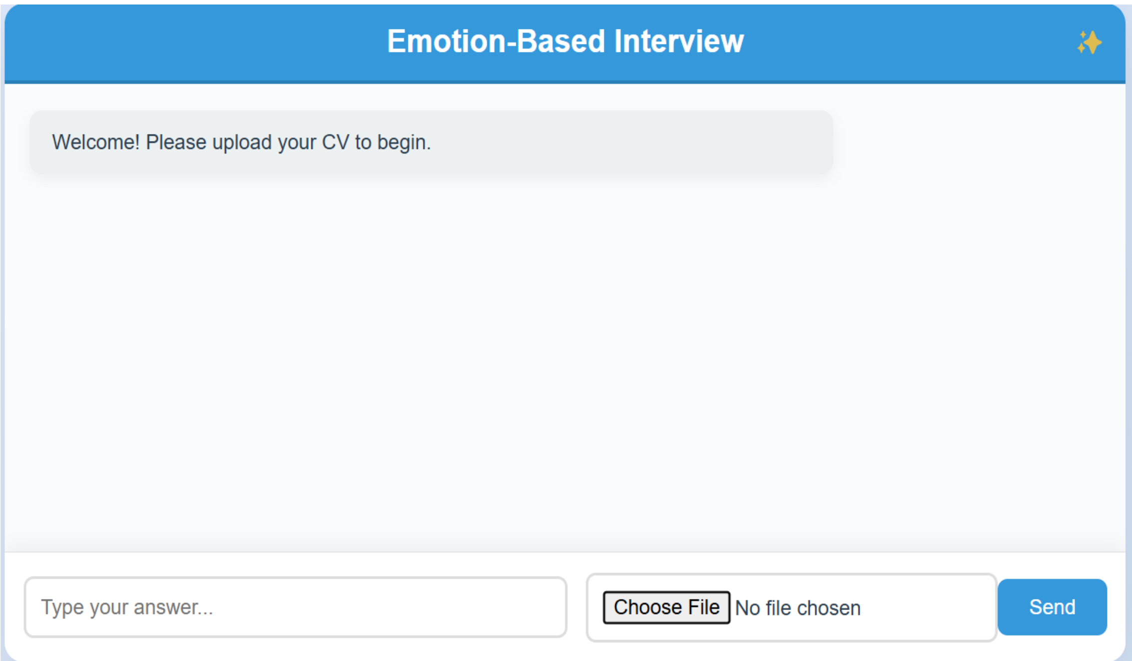


Figure 5.4.1 System's User Interface

To begin, the first image displays the system's user interface, which opens with a welcoming message: "Welcome! Please upload your CV to begin."

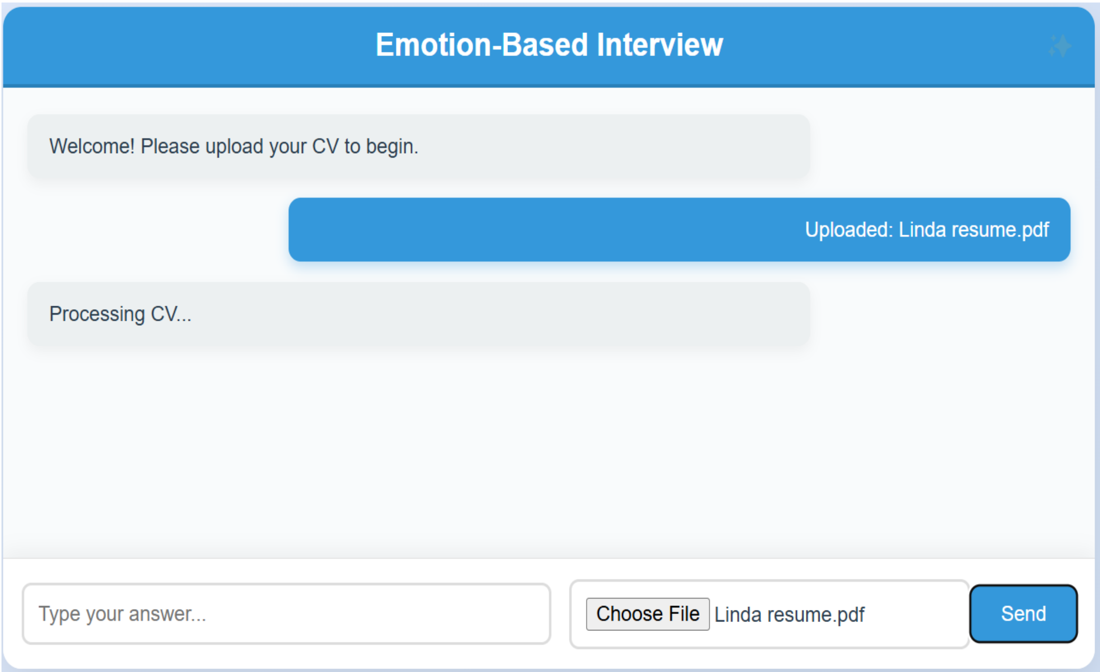


Figure 5.4.2 Processing CV

Figure 5.4.2 illustrates the moment when the CV is being uploaded. Once the upload is complete, the system displays a message prompt "Processing CV..." to indicate that it is analyzing the document.

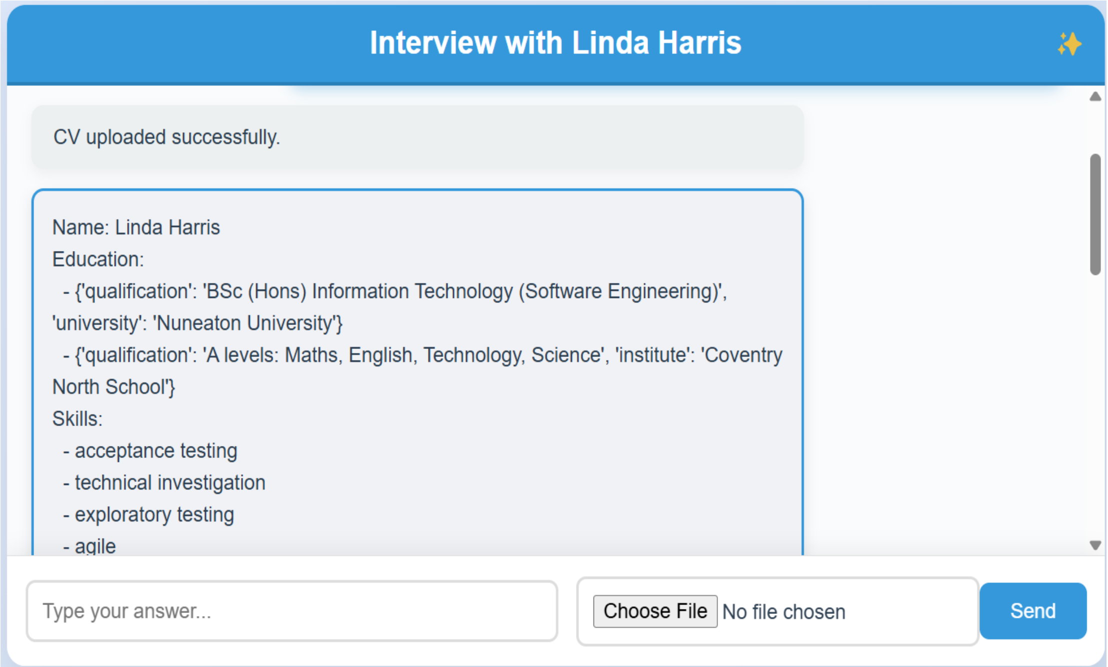


Figure 5.4.3 CV Uploaded Successfully

After the resume analysis is completed, the system notifies the user with a confirmation message stating, "CV uploaded successfully." This message indicates that the document has been thoroughly scanned and that key information, such as the candidate's full name, educational background, relevant skills, work experiences, and other critical details, has been successfully extracted. This extracted data is then displayed in a structured format on the interface for the user to review. Presenting this information allows both the system and the candidate to ensure that the correct data has been captured before moving forward with the interview.

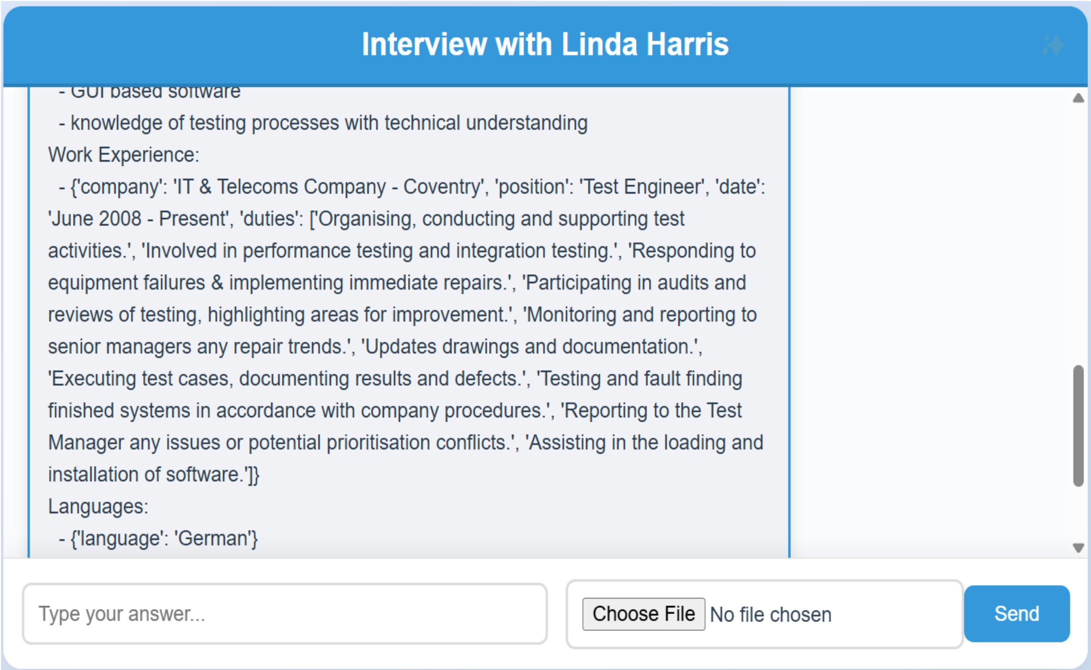


Figure 5.4.4 Details of CV

Next, as shown in the continuation from Figure 5.4.4, the system interface dynamically updates to reflect a more personalized environment. The main title of the system automatically changes to the name of the candidate, effectively transforming the generic interface into a personalized interview space. This transition not only enhances the user experience by adding a sense of professionalism and engagement but also marks the official beginning of the tailored AI-driven interview process. From this point onward, all questions and system responses will be based on the content of the uploaded CV, making the interview session uniquely relevant to the candidate’s background.

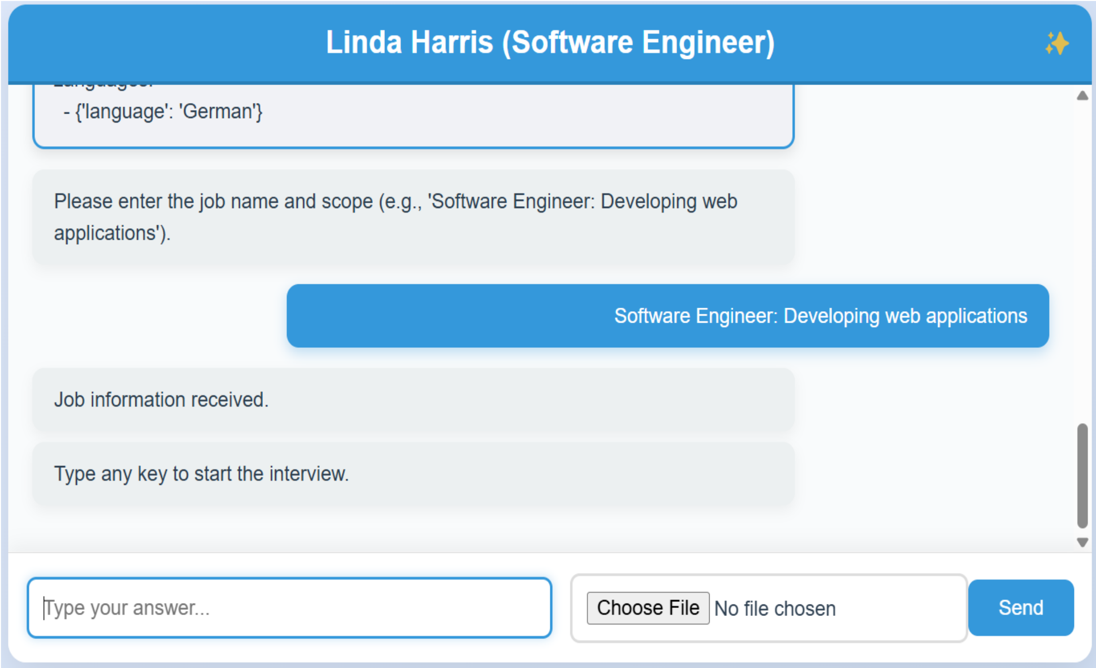


Figure 5.4.5 Input the Job Scope

In the next stage, the user is given the option to input the desired job scope or position they are applying for. This allows the system to further tailor the interview questions to match the requirements and expectations of that specific role. Once the job scope is entered, the system processes the input and returns a confirmation message such as “Job information received,” indicating that the relevant job details have been successfully loaded into the system. This step ensures that both the candidate's resume data and the job requirements are aligned for a more focused and accurate interview experience. With both the CV and job information now in place, the system displays a prompt indicating that everything is ready. At this point, the user is instructed to press any key to officially begin the interview session. This marks the transition from the setup phase to the interactive interview process, where the AI will start asking questions based on the candidate’s background and the selected job scope. At the same time, the system interface is updated to reflect the candidate’s full name along with the job position they are interviewing for, for example: “Interview with John Tan – Software Engineer.”

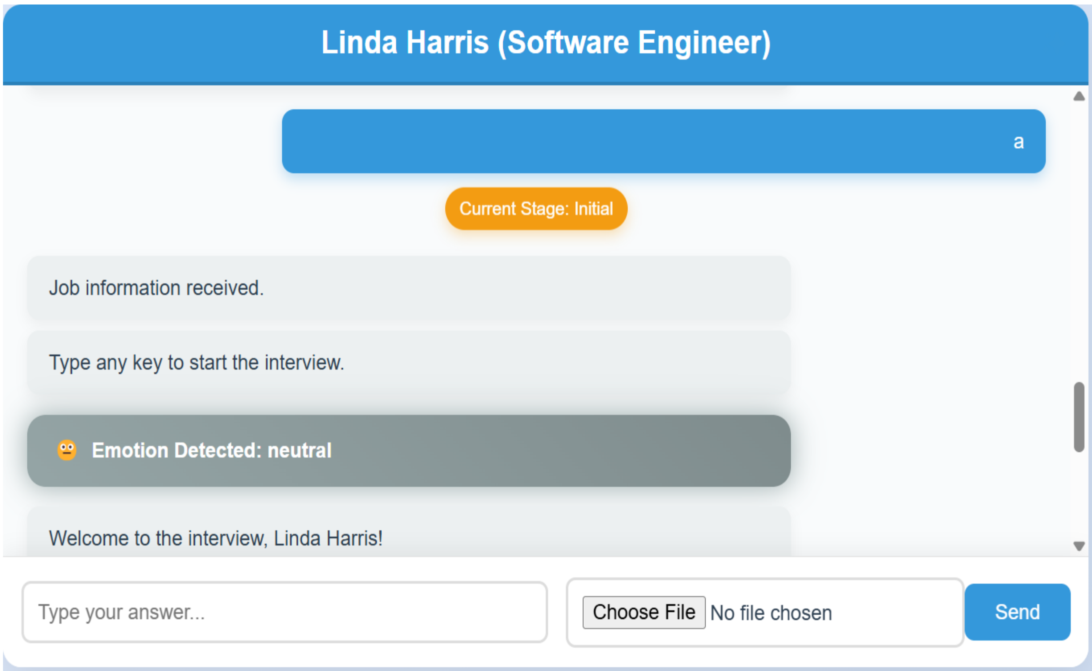


Figure 5.4.6 Initial Stage Started and First Emotion Detected

The first stage of the interview process is known as the initial stage. Before any questions are generated or asked, the system begins by activating its emotion detection module. At this point, the candidate's facial expressions are monitored in real-time using the system's built-in camera and computer vision capabilities. This initial emotional assessment allows the system to capture the candidate's baseline emotional state, such as happy, sad, neutral, angry and disgust before the formal questioning begins. By detecting emotions at the start, the system can adapt its tone and question difficulty more appropriately to the candidate's current state, creating a smoother and more human-like interview experience. Only after this preliminary emotion reading is complete does the system proceed to generate the first interview question, ensuring that the interaction begins with a thoughtful understanding of the candidate's non-verbal cues.

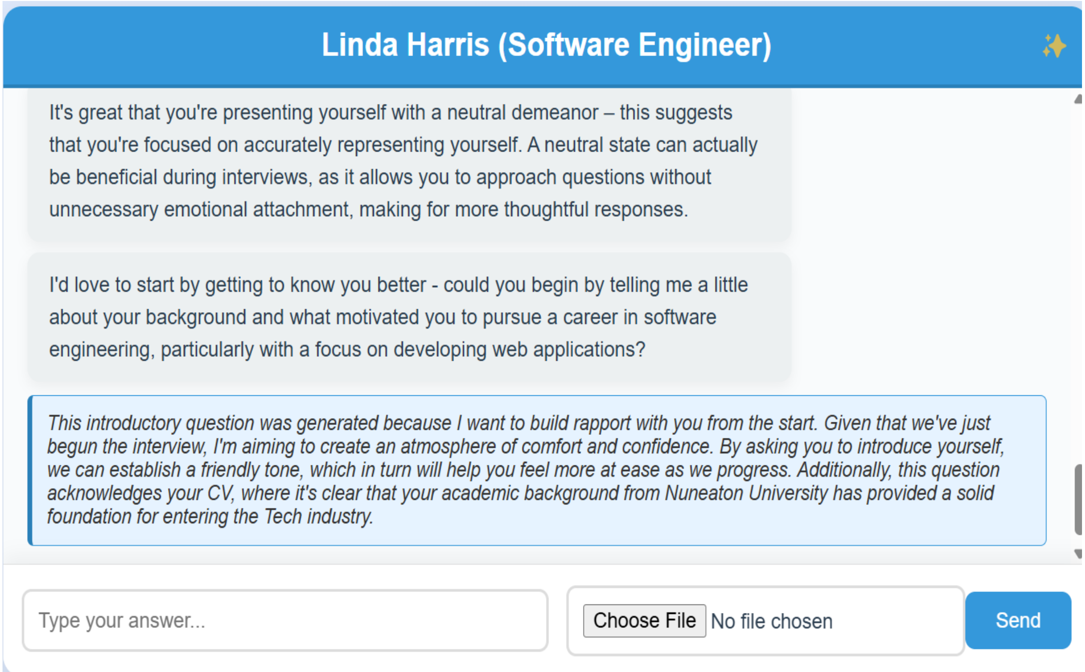


Figure 5.4.7 Questions and Explanations are Generated

Once the system completes the initial emotion detection, it proceeds to generate the first question. However, before diving directly into the questioning, the system begins by greeting the candidate in a way that reflects and responds to their detected emotional state. If the candidate appears fearful, or sad, the system will provide a calming and reassuring message to help ease their anxiety and create a more comfortable environment. On the other hand, if the candidate shows signs of happiness or excitement, the system responds with an uplifting and encouraging message to maintain a positive atmosphere and reinforce their confidence. This personalized emotional engagement ensures that the candidate feels acknowledged and supported right from the start of the interview.

Following the greeting, the system generates the first interview question based on the information extracted from the resume and the job scope previously provided. This question is not randomly selected, but rather carefully crafted to align with the candidate's background and the specific role being applied for. Just below the question, a blue box section appears on the interface. This section provides a clear explanation of the reasoning behind the generated question, such as which part of the resume it relates to, what skill or experience it aims to evaluate, or how it ties into the job requirements.

This added layer of transparency helps both the interviewer understand the logic behind the interview flow and showcases the AI’s ability to conduct context-aware questioning.

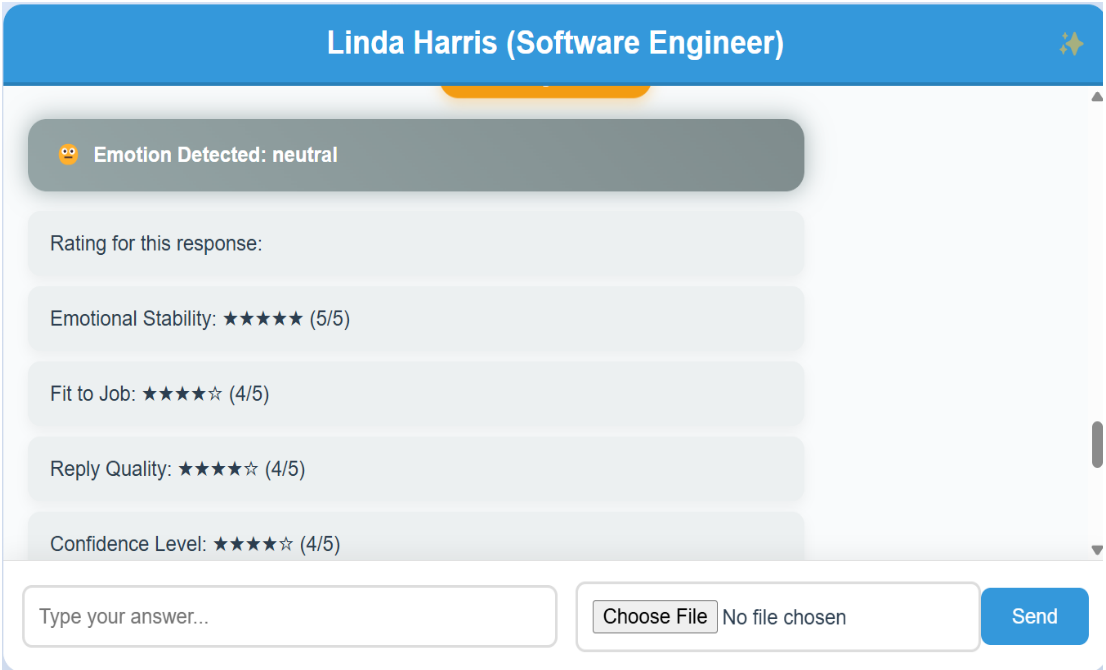


Figure 5.4.8 Questions with Explanations Generated (Initial Stage)

After the candidate responds to the initial question, the system transitions into the second stage of the interview process, known as the *Personal Stage*. Similar to the first stage, the system begins by detecting the candidate’s emotional state before proceeding further. This real-time emotion analysis plays a key role in understanding how the candidate is reacting to the interview environment and how their emotional stability evolves throughout the session.

Once the emotion is captured, the system evaluates the candidate’s response to the previous question using multiple criteria. These include **emotional stability** (how consistent the candidate's emotional state remained during the reply), **fit to job** (how well the response aligns with the job scope and requirements), **reply quality** (the clarity, relevance, and depth of the answer), and **confidence level** (inferred from facial expressions). This multi-dimensional assessment allows the system to provide a more holistic view of the candidate's performance, combining both verbal content and emotional cues. The results of this evaluation may be displayed in visual form, such as progress bars or ratings, offering insightful feedback that can be used by both interviewers and candidates to reflect on each stage of the interview.

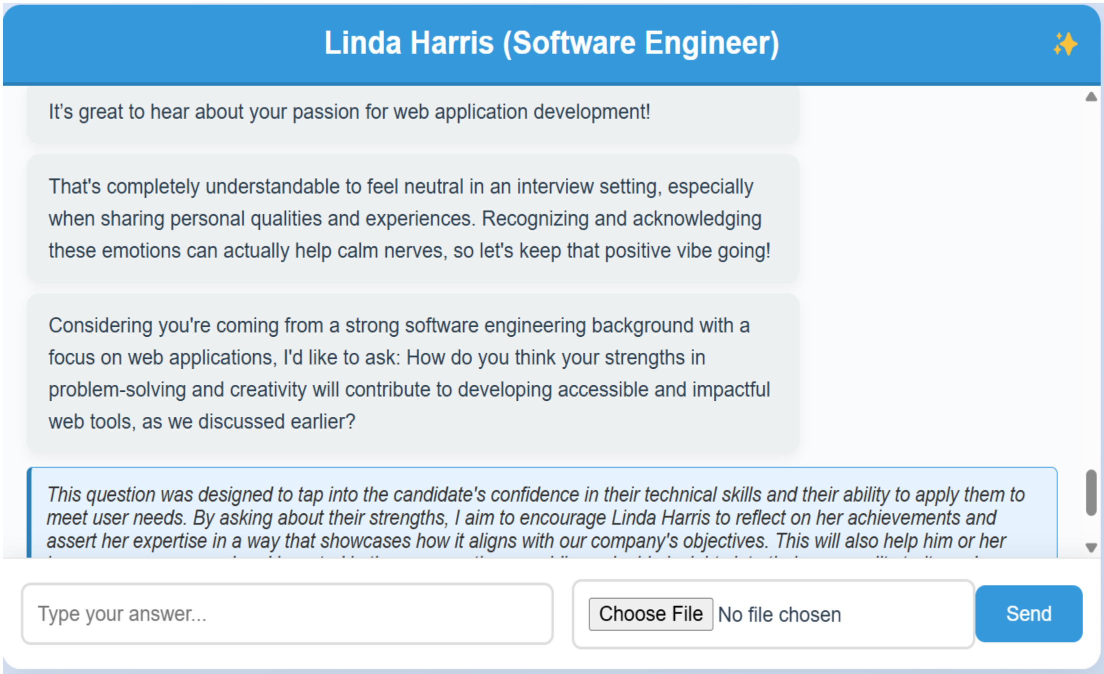


Figure 5.4.9 Questions with Explanations Generated (Personal Stage)

Figure 5.4.9 presents the generated questions and corresponding explanations for the *Personal Stage* of the interview. While the overall structure remains consistent with the previous *Initial Stage*, featuring the question at the top and a blue column below providing a detailed explanation of the content of the questions shifts significantly. In this stage, the questions are more personalized and focused on the candidate's individual experiences, motivations, values, and personality traits. They are designed to dig deeper into who the candidate is beyond their technical qualifications, helping to assess cultural fit and long-term potential for the role. The explanation below each question continues to clarify how the question was derived and its relevance to the candidate's background and the applied job role, ensuring full transparency in the AI's reasoning process.

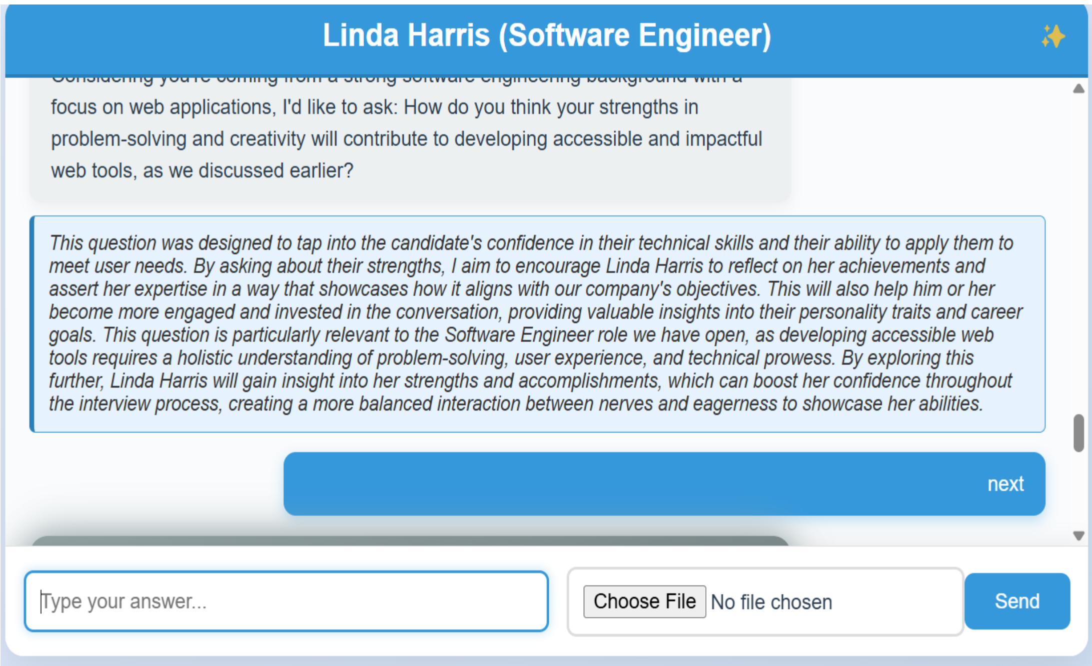


Figure 5.4.10 Full Explanations (Personal Stage) and “next” Function

Figure 5.4.10 displays the full explanation corresponding to the question shown in Figure 5.4.9, offering a more detailed breakdown of how the AI formulated the question and why it is relevant to the candidate’s background and the job scope. Additionally, before the candidate responds to the question, they can proceed to the next part of the interview by simply typing the command “next.” This user input triggers the system to transition seamlessly to the following section, ensuring a smooth and interactive experience throughout the multi-stage interview process.

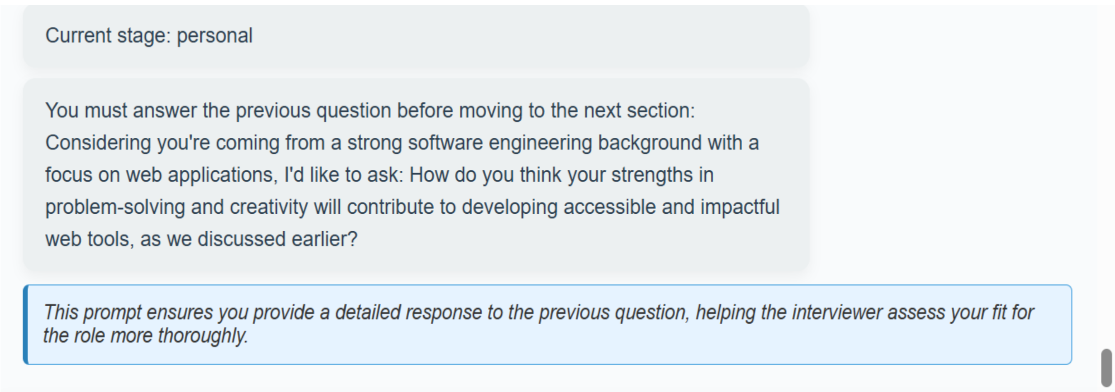


Figure 5.4.11 Require Answer from User

Figure 5.4.11 illustrates the prompt where the system requests the user to input their answer after inputting "next." This step emphasizes that the candidate must first provide a complete response to the previous question before the system can proceed to the next section. The system is designed to ensure that each answer is acknowledged and analyzed before generating any follow-up questions. Unlike simple sequential questioning, the follow-up question generated in the next section is not a direct continuation but belongs to a new interview segment with a different focus. However, the system still takes into account the candidate's previous responses to craft relevant and coherent questions that maintain contextual awareness. This approach ensures a structured yet dynamic interview flow that adapts to both the content and quality of the candidate's input.

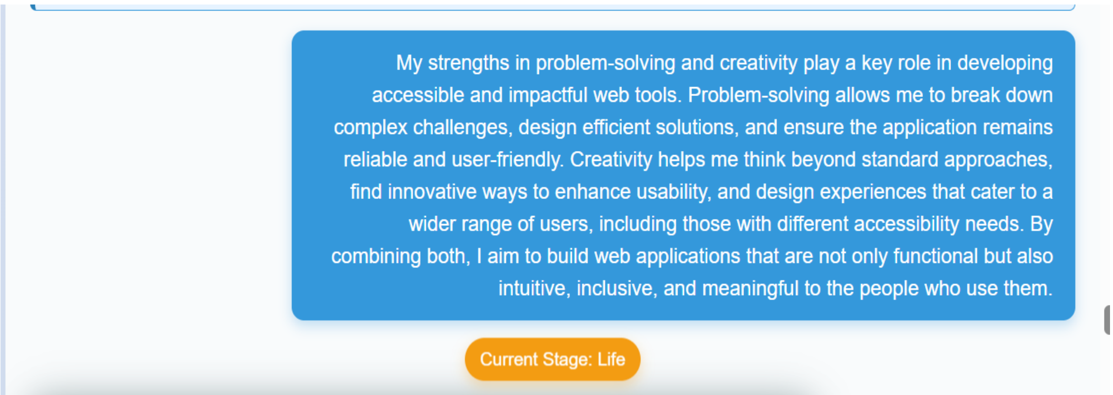


Figure 5.4.12 The Sample Input from User and Proceed to Next Section

Figure 5.4.12 displays a sample answer provided by the user and demonstrates the system's transition to the next section, known as the *Life Section*. Once the user submits their response, the system evaluates it and moves forward, continuing the interview journey. The *Life Section* shifts the focus toward understanding the candidate's personal background, lifestyle choices, values, and life experiences that may influence their working style and compatibility with the company culture.

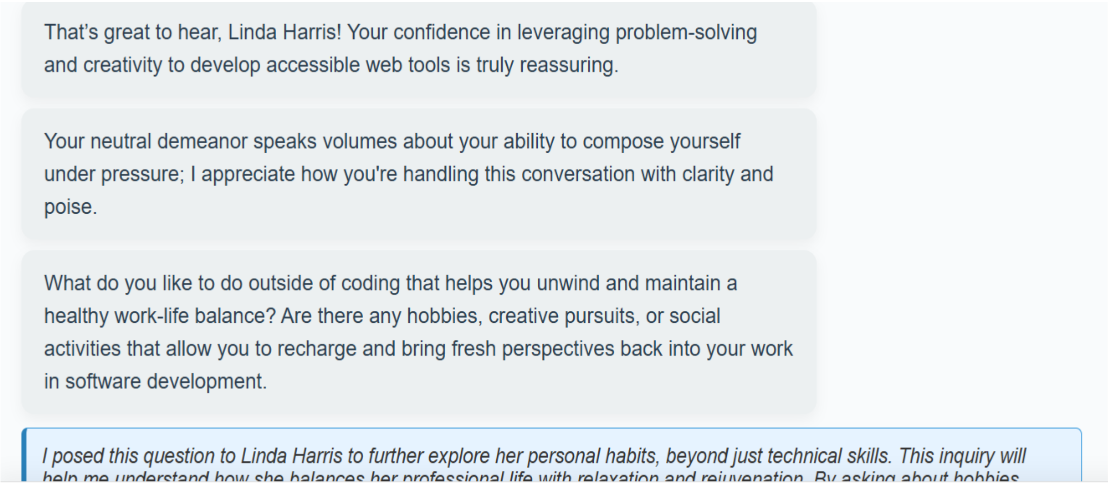


Figure 5.4.13 Questions with Explanations Generated (Life Stage)

Figure 5.4.13 showcases the questions and explanations generated during the *Life Stage* of the interview. While the visual structure remains consistent with previous stages, featuring the question at the top and a detailed explanation below in the blue-highlighted section. In this stage, the system generates questions that are more personal and reflective, focusing on the candidate's life experiences, values, daily habits, and how they approach challenges outside of a professional setting. These questions are designed to provide deeper insight into the candidate's personality and overall character, offering a deep understanding that goes beyond job skills and qualifications.

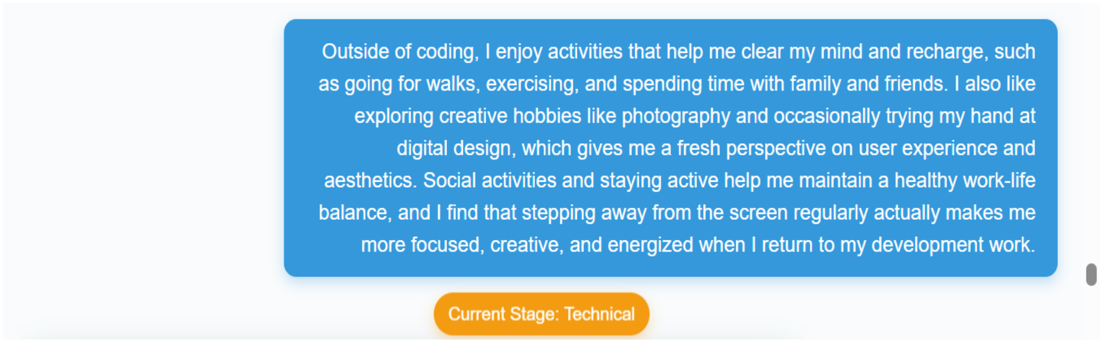


Figure 5.4.14 The Sample Input from User and Proceed to Next Section

Figure 5.4.14 presents a sample input provided by the user in response to a *Life Stage* question and demonstrates the system's transition to the next phase of the interview: the *Technical Stage*. The transition into the *Technical Stage* marks a shift in focus—from personal and experiential insights to assessing the candidate's domain knowledge, problem-solving skills, and technical expertise relevant to the job role. This staged approach not only keeps the interview organized but also enhances the depth and relevance of the evaluation process.

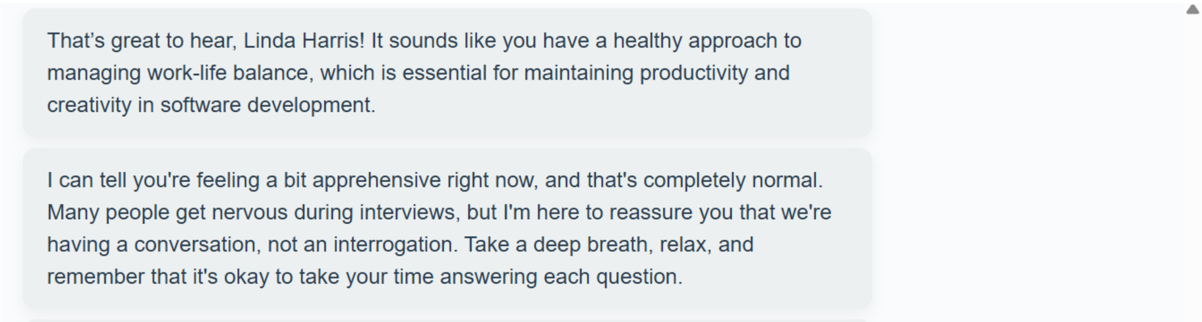


Figure 5.4.15 Questions Generated (Technical Stage)

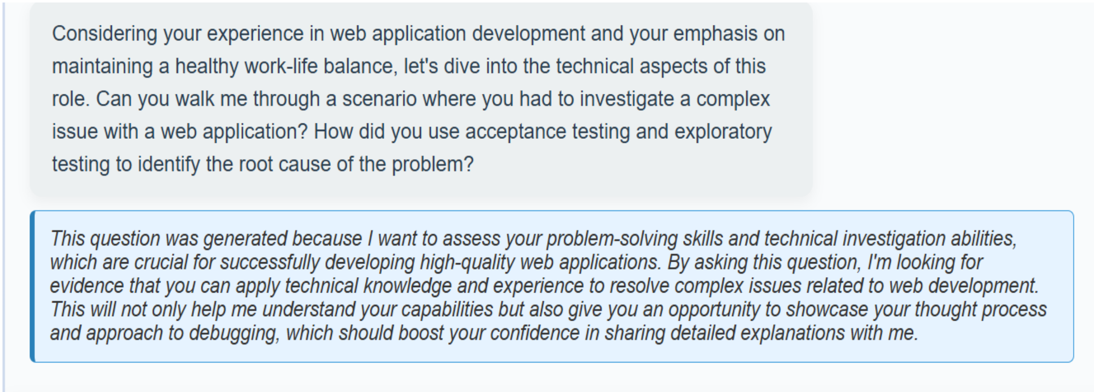


Figure 5.4.16 Questions with Explanations Generated (Technical Stage)

Figure 5.4.15 and 5.4.16 displays the questions and corresponding explanations generated during the *Technical Stage* of the interview process. Maintaining the same interface structure as in previous stages, the question appears at the top, followed by a detailed explanation in the blue-highlighted section beneath it. However, the content of the questions in this stage shifts to focus specifically on technical knowledge and job-related skills. These questions are tailored based on the candidate's resume, job scope, and any prior responses, ensuring that the interview remains personalized and relevant. The goal of this stage is to evaluate the candidate's proficiency in their field, including their understanding of key concepts, problem-solving abilities, and practical experience. The explanation section continues to provide transparency into how and why the question was generated, helping both the candidate and evaluator understand the AI's reasoning and its alignment with the role's technical requirements.

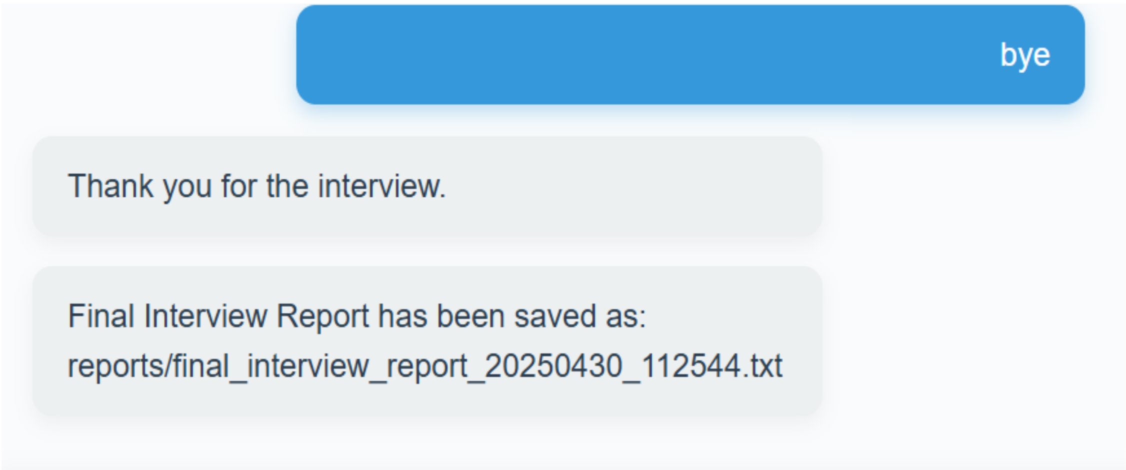


Figure 5.4.17 End of Interview

Figure 5.4.17 illustrates the final step of the interview process, where the user decides to end the session by typing the command "bye." Once this command is entered, the system automatically closes the camera, signaling the end of the real-time emotion detection and interaction. At the same time, the system compiles all collected data, including the user's answers, detected emotions, stage-by-stage evaluations, and question explanations and generates a comprehensive final report. This report is then saved into a designated folder for future reference and further evaluation by recruiters or analysts.

```
File    Edit    View

Interview Evaluation Report
=========================
Overall Emotional Stability: ★☆☆☆☆ (1/5)
Overall Fit to Job Requirements: ★★☆☆☆ (2/5)
Overall Reply Quality: ★☆☆☆☆ (1/5)
Overall Confidence Level: ★☆☆☆☆ (1/5)
-------------------------
Overall Rating: ★☆☆☆☆ (1/5)
Fit Probability: 20%
```

Figure 5.4.18 Details of Interview Evaluation Report

The interview evaluation report generated by the system is divided into two main sections, each offering valuable insights into the candidate's performance. The first section provides a comprehensive breakdown of four key evaluation metrics: overall emotional stability, overall fit to job requirements, overall reply quality, and overall confidence level. Each of these aspects is rated on a scale from 1 to 5. For every interview question, the system detects the candidate's emotional state and evaluates the provided answer across all four metrics, assigning individual scores for each. These scores are calculated using a combination of facial expression analysis, natural language processing, and relevance to the job role. As the interview progresses, these scores are recorded and accumulated, and by the end of the session, the system computes the overall rating for each metric by averaging the scores from all the questions answered.

The second section of the report summarizes the candidate's overall performance by presenting the mean score of all four-evaluation metrics. In addition to the average rating, this section includes a fit probability percentage, which estimates how suitable the candidate is for the job. This percentage is generated based on a combination of emotional trends, communication quality, and the degree of alignment between the candidate's profile and the job requirements. Together, both sections provide a structured, data-driven summary to support informed decision-making in the recruitment process.

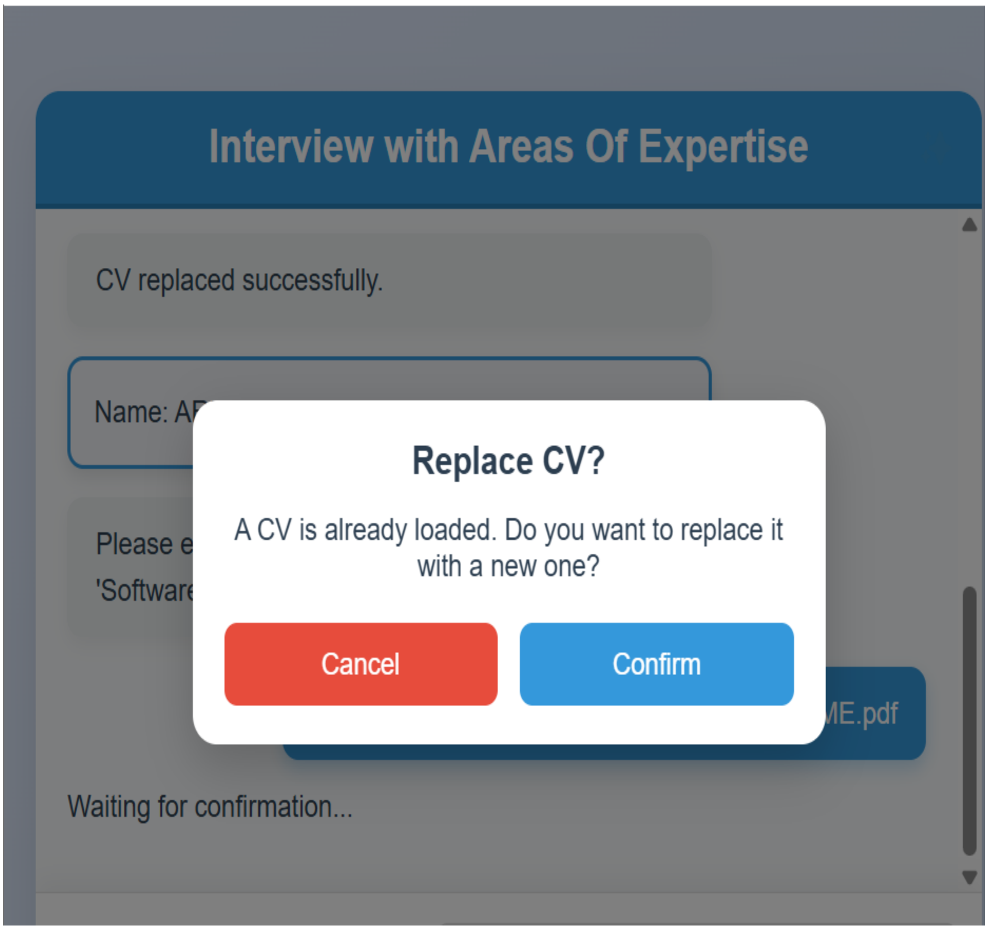


Figure 5.4.19 Replace a New CV

Figure 5.4.19 illustrates the confirmation prompt that appears when a candidate chooses to upload a new CV. This feature is an additional feature and designed to provide users with smooth and user-friendly experience by allowing them to easily replace an incorrectly uploaded resume without any complications or hesitation. Upon triggering the upload process again, the system detects the action and displays a confirmation message to ensure that the user truly intends to overwrite the previously uploaded CV. This additional step helps prevent accidental replacements while offering flexibility to correct mistakes early in the process. By supporting this kind of error handling, the system enhances usability and ensures that the interview is conducted with the correct and intended resume content.

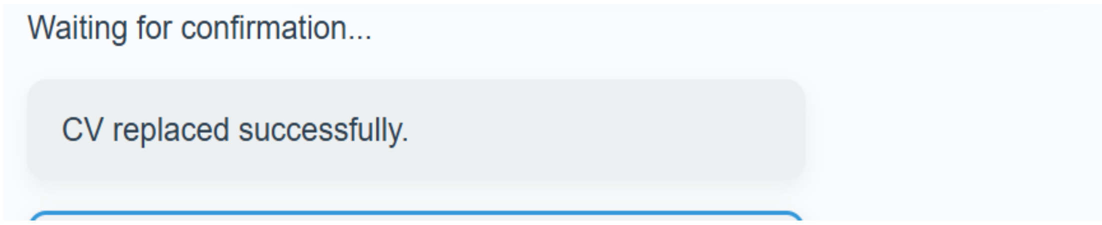


Figure 5.4.20 Confirmation of Successful Replace CV

Figure 5.4.20 shows the confirmation message that appears once the CV has been successfully replaced. This message serves as a clear indication to the user that the system has accepted and processed the new resume file. It reassures candidates that the previously uploaded CV has been overwritten and that the new one will now be used for analysis and interview preparation. By providing this confirmation, the system ensures transparency and eliminates any confusion about whether the replacement was successful. This feature adds to the overall user experience by making the process intuitive, efficient, and reliable, especially in situations where users may have uploaded an incorrect file initially.

5.5 Implementation Issues and Challenges

The development of the InterviewAI system presented a range of formidable challenges that significantly influenced its implementation and overall performance. A primary obstacle was the complex integration of multiple software libraries and technologies into a unified system, which introduced substantial compatibility issues that required careful resolution. Libraries such as PyPDF2, TensorFlow, OpenCV, and Flask, each integral to critical functionalities like CV parsing, emotion detection, question generation, and interface development often came with specific version dependencies on Python or underlying frameworks, leading to frequent conflicts. For example, a library like TensorFlow might require an older version to function optimally with certain dependencies, whereas the Llama 3.2 Vision model, used for CV analysis and question generation, necessitates TensorFlow 2.10 or PyTorch 1.13, creating a version mismatch that resulted in errors or system crashes during execution. Moreover, these components were typically developed in disparate environments, leading to inconsistencies in data structures, formats, or processing methods. A notable instance was the differing image formats between OpenCV, which processes images in BGR format, and the Emotion Detection Model’s CNN, which expects RGB input, necessitating additional conversion steps and extensive adjustments to ensure smooth interoperability across the system’s modules, ultimately demanding significant time and effort to achieve stable integration.

Another substantial challenge was the variability in CV formats submitted by candidates, which posed considerable difficulties in achieving accurate and reliable data extraction. CVs presented in formats other than PDF, such as Word documents, scanned images, or even handwritten resumes, were particularly problematic due to their inconsistent layouts, varied fonts, embedded graphics, or non-standard structures, often leading to incomplete or erroneous extraction of critical information like education, work experience, or skills. For instance, a Word document with complex tables or a scanned image with low resolution might result in missing key details, such as the candidate’s degree or job titles, thereby affecting the quality of the interview questions generated. After thoroughly evaluating various formats, PDF was identified as the most consistent and reliable option for parsing, leveraging libraries like PyPDF2 for text extraction and pdfplumber for handling complex layouts. However, this limitation significantly restricted the system’s flexibility, as candidates submitting CVs in non-PDF formats required manual conversion to PDF before processing, a step that introduced delays in the interview preparation process and increased the workload for the HR Professional. This constraint underscored the need for a more robust and versatile parsing mechanism capable of handling a diverse array of CV formats without compromising the accuracy or efficiency of data extraction, highlighting a critical area for future improvement to enhance the system’s usability and accessibility for a broader range of candidates.

5.6 Concluding Remark

This chapter has outlined the implementation process of the InterviewAI system, providing a comprehensive guide to its setup and operation while addressing the challenges faced. Section 5.1 showed users how to set up the hardware for the project, detailing the necessary equipment to ensure the system runs effectively. Section 5.2 explained what software users need and how to implement it, covering the installation of required libraries and frameworks to support the system's functionality. Section 5.3 described the settings and configuration process, guiding users on how to configure the system for optimal performance. Section 5.4 demonstrated the system's operation with screenshots, illustrating how it functions during an interview, from question generation to response recording. Finally, Section 5.5 highlighted implementation issues and challenges, such as software compatibility conflicts, and variability in CV formats submitted by candidates. Together, these sections provide a clear roadmap for implementing the InterviewAI system, overcoming obstacles, and preparing for its evaluation in the next chapter, where its real-world performance will be thoroughly assessed.

Chapter 6 System Implementation

6.1 System Testing and Performance Metrics

Emotion Detection Model

The Emotion Detection Model within the InterviewAI system, a Convolutional Neural Network (CNN) trained on the RAF-DB dataset, underwent an performance evaluation to confirm its effectiveness in real-time emotion recognition, a critical component for adapting interview questions to the candidate's emotional state across the four sections—Initial, Personal, Life, and Technical. This evaluation encompasses a detailed analysis of training and testing metrics, loss and accuracy curves over the training period, and a comprehensive classification report, providing a thorough understanding of the model's reliability in identifying emotions such as surprise, fear, disgust, happy, sad, angry, and neutral.

During the training phase, the model was assessed over 575 batches, achieving a train accuracy of 91.30% and a train loss of 24.20%. This high training accuracy demonstrates the model's strong capability to correctly classify emotions on the training data, while the relatively low loss indicates that the model's predictions closely align with the true labels, minimizing errors during the learning process. The training was conducted over 60 epochs, and the loss curve provides further insight into the model's learning behavior. The curve shows both train and validation loss starting at approximately 1.75 and steadily decreasing, with the train loss converging to around 0.25 and the validation loss stabilizing at approximately 0.50 by the end of training. This consistent decline in loss, without significant divergence between the train and validation curves, suggests that the model learned effectively without overfitting, maintaining its ability to generalize to new data. Complementing the loss curve, the accuracy curve illustrates a rapid improvement in performance, with the train accuracy climbing from 0.40 to 0.91 and the validation accuracy rising from 0.40 to 0.83, both stabilizing after approximately 30 epochs.

For the testing phase, the model was evaluated over 192 batches, completing the assessment in 17 seconds at 86 ms per step, resulting in a test accuracy of 83.04% and a test loss of 52.68%. While the test accuracy is slightly lower than the training accuracy, it remains sufficiently high for practical application in a real-world interview setting, ensuring that the system can reliably detect emotions. The higher test loss, compared to the training loss, suggests some challenges in generalizing to unseen data, potentially due to factors such as inconsistent lighting, diverse facial angles, or partial occlusions, which were identified as challenges during development. Despite this, the test accuracy of 83.04% confirms the model's ability to perform effectively under typical conditions, supporting the InterviewAI system's requirement for real-time emotion recognition to adapt questions dynamically, such as simplifying questions for a nervous candidate to create a more supportive interview environment.

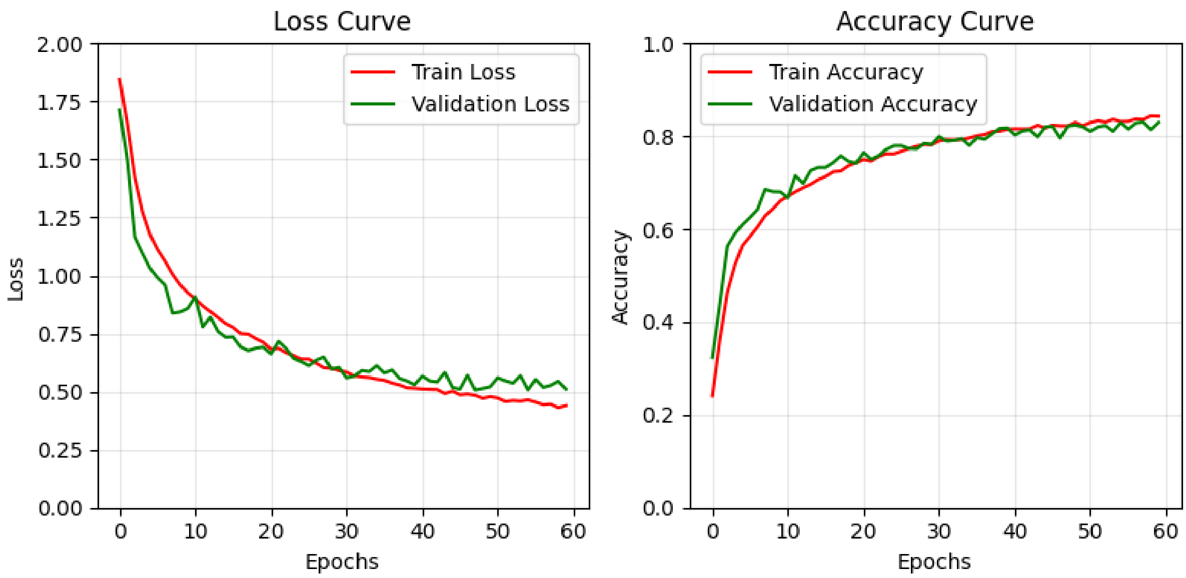


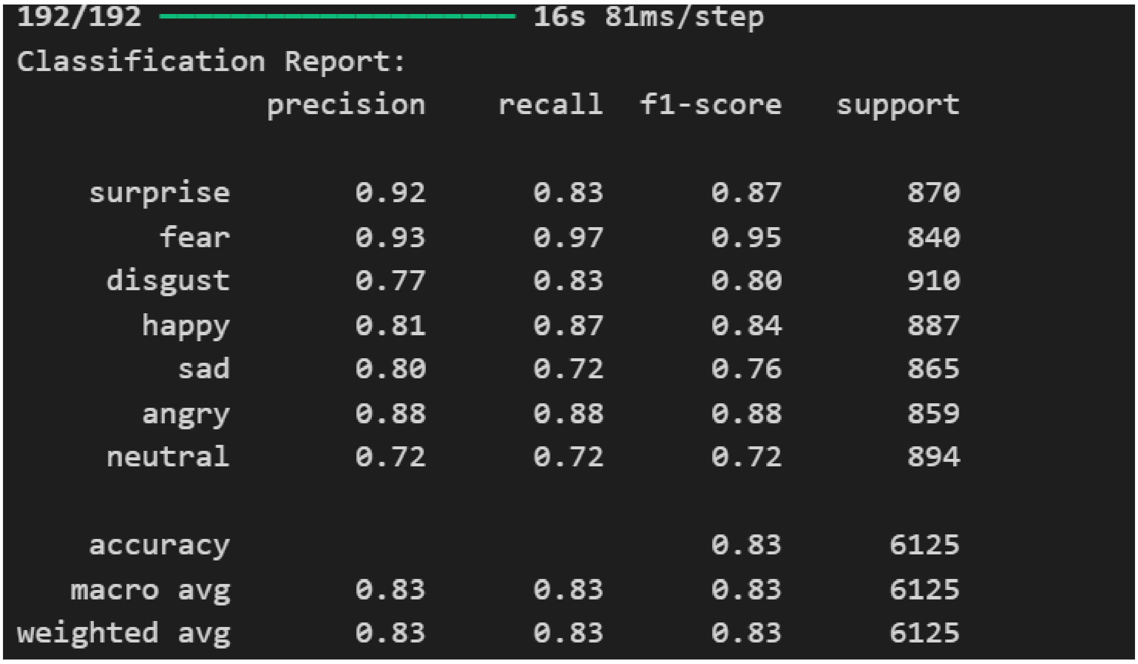
Figure 6.1.1 Graph of Loss Curve and Accuracy Curve

```
575/575 ━━━━━━━━━━━━━━━━━━━━ 43s 73ms/step - accuracy: 0.9146 - loss: 0.2433
192/192 ━━━━━━━━━━━━━━━━━━━━ 17s 86ms/step - accuracy: 0.8365 - loss: 0.4977
Train Loss: 24.20%
Train Accuracy: 91.30%

Test Loss: 52.68%
Test Accuracy: 83.04%
```

Figure 6.1.2 Details of Train and Test Accuracy and Loss

A detailed classification report showed I Figure 6.2.2 provides a deeper analysis of the model's performance across the seven emotion classes for a test set of 6,125 samples. The report reveals balanced overall performance, with a macro average and weighted average precision, recall, and F1-score of 0.83, indicating consistent performance across all classes. Specific emotion classes demonstrated varying levels of accuracy: fear' achieved the highest F1-score of 0.95 (precision: 0.93, recall: 0.97, support: 840), reflecting the model's strong ability to detect this emotion, followed by 'angry' with an F1-score of 0.88 (precision: 0.88, recall: 0.88, support: 859), indicating reliable detection for more distinct emotional expressions. The 'surprise' class also performed well, with an F1-score of 0.87 (precision: 0.92, recall: 0.83, support: 870), while 'happy' scored an F1-score of 0.84 (precision: 0.81, recall: 0.87, support: 887), showing good detection for positive emotions. However, the model exhibited slightly lower performance for more subtle emotions, with 'disgust' achieving an F1-score of 0.80 (precision: 0.77, recall: 0.83, support: 910), 'sad' at 0.76 (precision: 0.80, recall: 0.72, support: 865), and 'neutral' at 0.72 (precision: 0.72, recall: 0.72, support: 894). Despite these variations, the overall accuracy of 0.83 across all classes demonstrates the model's capability to support the InterviewAI system's objective of real-time emotion recognition, enabling the system to adapt questions effectively based on the Candidate's emotional state, thus enhancing the overall interview experience.


192/192 ━━━━━━━━━━━━━━━━━━━━ 16s 81ms/step
Classification Report:

	precision	recall	f1-score	support
surprise	0.92	0.83	0.87	870
fear	0.93	0.97	0.95	840
disgust	0.77	0.83	0.80	910
happy	0.81	0.87	0.84	887
sad	0.80	0.72	0.76	865
angry	0.88	0.88	0.88	859
neutral	0.72	0.72	0.72	894
accuracy			0.83	6125
macro avg	0.83	0.83	0.83	6125
weighted avg	0.83	0.83	0.83	6125

Figure 6.1.3 Classification Report of Emotion Model

The Emotion Detection Model, a Convolutional Neural Network (CNN) trained on the RAF-DB dataset, proficiently categorizes emotions such as surprise, fear, disgust, happy, sad, angry, and neutral, as demonstrated through comprehensive testing across these seven distinct classes.

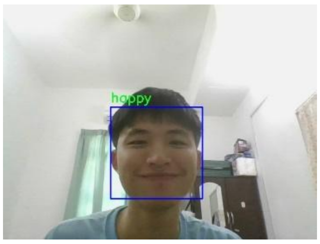


Figure 6.1.4 Emotion detected as 'happy'

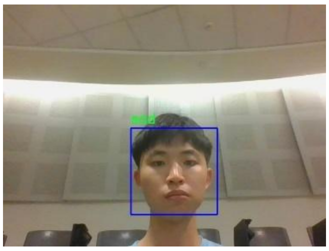


Figure 6.1.5 Emotion detected as 'sad'

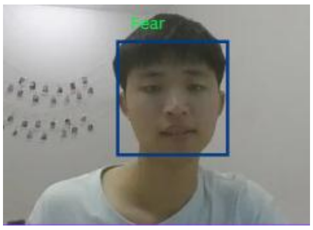


Figure 6.1.6 Emotion detected as 'fear'

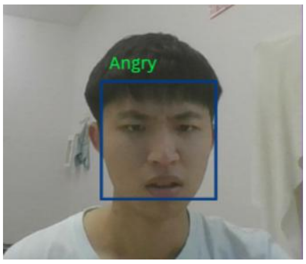


Figure 6.1.7 Emotion detected as 'angry'

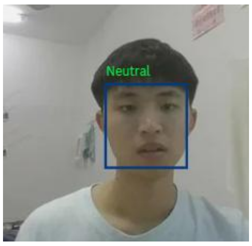


Figure 6.1.8 Emotion detected as 'neutral'

Figure 6.1.9 Emotion detected as 'disgust'

Question Generation Model

The Question Generation Model is an important part of the InterviewAI system, designed to use natural language processing (NLP) to create personalized and relevant interview questions. It uses Llama 3, a large language model (LLM) from Ollama, to perform its main tasks. The model generates initial and follow-up questions based on the Candidate's detected emotions and their background information extracted from their CV. To test how emotions affect the questions, we ran experiments in two ways:

one with emotion detection and one without, to compare the differences in the generated questions, as shown in the table below.

Emotion	Question Generated Based on Emotion
Without Emotion	"Can you tell me a little bit about yourself and your professional journey so far?"
Happy	"I see you're feeling great today! Can you share an achievement from your previous job that made you feel particularly proud?"
Sad	"I notice you seem a bit down. Is everything alright? Would you like to talk about any challenges you've faced recently in your professional life?"
Angry	"I sense some frustration. Can you tell me about a time when you faced a conflict at work and how you handled it?"
Disgust	"It seems like something may be bothering you. Can you describe a work environment or situation that you found challenging or uncomfortable?"
Surprise	"You look a bit surprised. Did something unexpected happen today? How do you usually handle surprises or changes in your work environment?"

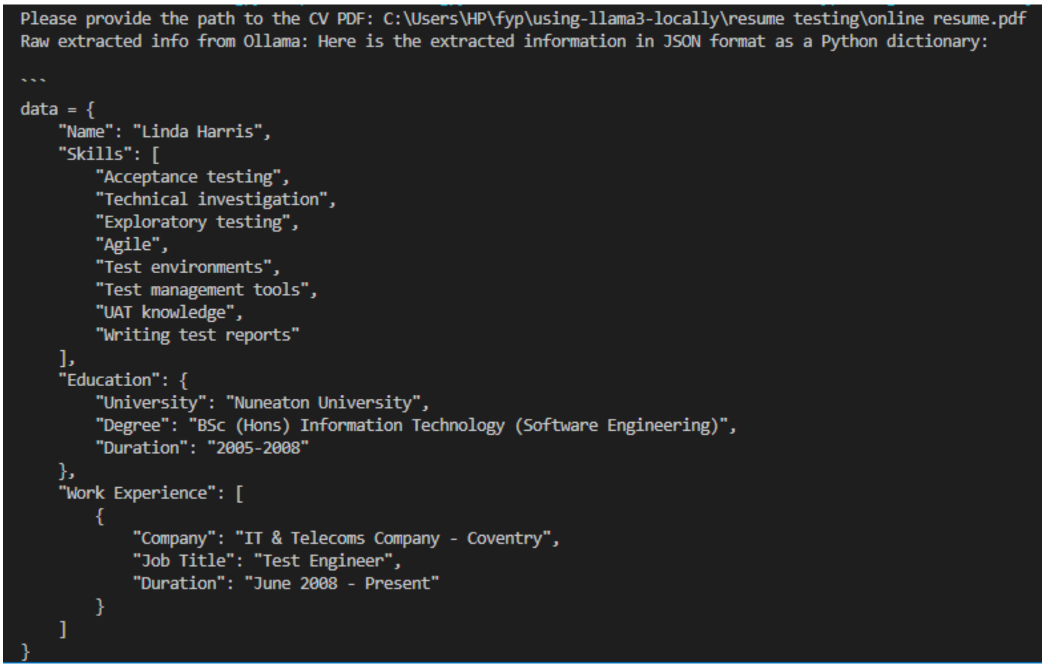
Neutral	"Let's start with your experience. Can you walk me through your most recent role and what your key responsibilities were?"
Fear	"You appear to be a bit nervous. Is there anything specific about this interview that worries you? How do you typically handle stressful situations at work?"

Table 6.1.1 Difference of Questions Generated Based on Emotion and Without Emotion

The results highlight a clear difference between questions generated with and without emotion detection in the InterviewAI system. When emotion detection is not used, the model produces a standard question like, "Can you tell me a little bit about yourself and your professional journey so far?" which is a formal and general way to ask for background information. In contrast, when emotion detection is applied, the questions adapt to the candidate's emotional state, making them more personalized and engaging. For example, if the candidate appears happy, the model generates a question such as, "I see you're feeling great today! Can you share an achievement from your previous job that made you feel particularly proud?" This approach boosts the candidate's energy and creates a more tailored experience by focusing on positive experiences. Similarly, if the candidate seems sad, the model responds with a question like, "I notice you seem a bit down. Is everything alright? Would you like to talk about any challenges you've faced recently in your professional life?" This question first addresses the candidate's emotional state to help them feel calm and supported before delving into their professional challenges. These emotion-based questions make candidates feel more comfortable and understood, enhancing the overall interview experience.

Resume Analysis Model

The initial development of the Resume Analysis Model centers on establishing a streamlined process for automatically extracting essential information from resumes to support the InterviewAI system's question generation. This process employs PyMuPDF to parse text from PDF documents, ensuring accurate retrieval of all relevant content despite variations in resume layouts or structures. Once the text is extracted, the system identifies key details from the PDFs, which are then processed and forwarded to the Large Language Model (LLM), LLaMA 3. This model conducts an in-depth analysis of the extracted text, systematically identifying and categorizing vital components such as the candidate's name, university, degree, skills, work experience, and other pertinent qualifications. Through the integration of these technologies, the Resume Analysis Model is finely tuned for PDF parsing, enabling efficient and precise extraction, processing, and analysis of resumes while delivering a thorough summary of a candidate's credentials with minimal manual effort. The figures below provide examples of the key information successfully extracted from resumes, illustrating the model's capability.



```
Please provide the path to the CV PDF: C:\Users\HP\fyp\using-llama3-locally\resume testing\online resume.pdf
Raw extracted info from Ollama: Here is the extracted information in JSON format as a Python dictionary:

```

data = {
    "Name": "Linda Harris",
    "Skills": [
        "Acceptance testing",
        "Technical investigation",
        "Exploratory testing",
        "Agile",
        "Test environments",
        "Test management tools",
        "UAT knowledge",
        "Writing test reports"
    ],
    "Education": {
        "University": "Nuneaton University",
        "Degree": "BSc (Hons) Information Technology (Software Engineering)",
        "Duration": "2005-2008"
    },
    "Work Experience": [
        {
            "Company": "IT & Telecoms Company - Coventry",
            "Job Title": "Test Engineer",
            "Duration": "June 2008 - Present"
        }
    ]
}
```

*Figure 6.1.10 Results of extracting CV (user 1)*
```

```
{
    "Name": "Weiloon Tan",
    "Skills": [
        {"CAD software": ["Solidworks", "Autocad"]},
        {"Programming Languages": ["C++", "Python"]},
        {"Microsoft Office": ["Word", "Excel", etc.]},
        {"Other Skills": ["Basic German", "Drawing Vacuum Cleaner using Solidworks"]}
    ],
    "Education": [
        {
            "University": "UNIVERSITI TUNKU ABDUL RAHMAN",
            "Degree": "Bachelor of Mechanical Engineering with Honours",
            "CGPA": 3.5557,
            "Degree": "Bachelor of Mechanical Engineering with Honours",
            "CGPA": 3.5557,
            "CGPA": 3.5557,
            "Year": "in progress (Year 3)"
        },
        {"High School": "CHUNG LING HIGH SCHOOL", "Stream": "Science Stream", "Years": "2015-2020"}
    ]
}
```


Figure 6.1.11 Results of extracting CV (user 2)


```
{
    "Name": "Soon Chun Hong",
    "Skills": [
        "Python",
        "SQL",
        "C++",
        "HTML",
        "Java Script",
        "CSS",
        "R",
        "Java"
    ],
    "Education": {
        "University": "Universiti Tunku Abdul Rahman (UTAR)",
        "Degree": "Bachelor of Computer Science (Hons)",
        "CGPA": 3.56
    }
}
```


Figure 6.1.12 Results of extracting CV (user 3)

6.2 Testing Setup and Result (AB test)

To evaluate the effectiveness of the InterviewAI system's emotion detection feature in question generation, an A/B testing approach was conducted with 10 interviewers and 10 interviewees. The test aimed to compare the interview experience with and without emotion detection, focusing on candidate engagement, comfort, and overall usability. Below is the detailed preparation, pairing, testing process, and a structured table of ratings and comments from participants, reflecting their experiences before and after using the emotion detection feature, including realistic feedback with some recommendations for improvement.

6.2.1 Testing Setup

The preparation for the A/B testing involved recruiting a diverse group of participants to ensure a wide range of perspectives. A total of 10 interviewers were selected, consisting of individuals from various departments who were either graduates, current employees, or team managers. For the interviewees, a group of 10 candidates, primarily students or recent graduates from different academic programs, were invited to participate in the testing process. The interviewers were given a detailed briefing on how to use the InterviewAI system's interface, which was built using Flask, HTML, and JavaScript, to view and ask questions generated by the system and to type the interviewees' verbal responses. The test spanned approximately 7 days to complete the entire evaluation process.

The 10 interviewers and 10 interviewees were paired for the A/B testing, with pairings arranged based on their availability to ensure scheduling compatibility. Each pair conducted two interview sessions: the first session utilized the InterviewAI system without emotion detection, where the model produced standard questions, and the second session enabled emotion detection, allowing the model to adjust questions according to the interviewee's emotional state, as identified by the CNN model trained on the RAF-DB dataset. The interviewers adhered to the system's generated questions, which were displayed on the interface, and recorded the interviewees' responses, maintaining consistency across all pairs.

Each pair participated in two interview sessions. The first session followed a traditional interview format, with the system generating standard questions without considering emotion detection. In the second session, the system customized questions based on the interviewee's detected emotional state, tailoring the interaction to their mood. After each session, both the interviewer and the interviewee rated their experience on a scale from 1 to 10 (1 being poor, 10 being excellent) and shared comments reflecting their experiences during the two sessions, including suggestions for improvement where relevant.

6.2.2 Testing Result

The table below summarizes the ratings and comments from both interviewers and interviewees, comparing their experiences before (without emotion detection) and after (with emotion detection) using the InterviewAI system.

Pair	Name (Position)	Rating (Without Emotion Detection Model)	Rating (With Emotion Detection Model)	Comment / Recommendation
1	Sally Wong (Senior HR Manager)	6	8	The first session's questions were too standard and didn't suit Shalini's mood, making it hard to connect. With emotional detection, it noticed she was happy and asked about her achievements, which made the session more engaging.
	Shalini a/p Ravi (HR Management Student)	5	7	The first session felt a bit formal; I was happy, but the questions didn't reflect that, so I didn't share much. The second session was better, it saw my mood and asked about something positive, which made me more comfortable.

2	Lee Chun Tat (R&D Team Lead)	5	9	Desmond was nervous in the first session, but the generic questions didn't help him relax, which felt awkward. With emotion detection, it asked if he was okay, helping him open up more. It's a great improvement.
	Desmond Ong (Computer Science Student)	4	9	I was really nervous in the first session, and the questions didn't make me feel better, it was too formal. The second session was much better. It noticed my nervousness and asked if I was okay, which calmed me down. I'd like the system to give more encouragement to boost my confidence.
3	Tan Yong Pang (Senior Mechanical Engineer)	6	8	The first session didn't adapt to Karthik's neutral mood, so I had to try harder to engage him. With emotion detection, it helps him to talk more. It's helpful, but typing is quite inconvenient for me.
	Karthik a/l Murugan (Mechanical	6	8	The questions asked in the second session are more creative, which engage me to share more to the interviewer without intension.

	Engineering Student)			
4	Ahmad Faiz bin Zainal (Bio Team Lead - Biomedical Research)	4	9	Lee Lei was happy, but the first session's formal questions didn't match her energy, making it feel dull. With emotion detection, it asked about her proud moments, which brought out her enthusiasm. It's a big improvement.
	Lee Lei (Bio Med Student)	5	10	I was feeling good in the first session, but the formal questions didn't let me show my excitement, it was boring. The second session was great, it noticed I was happy and asked about an achievement, which made it fun. Also, the system is nice as it asked many questions that related to my course!
5	Priya a/p Subramaniam (HR Lead - Recruitment)	5	8	Afiqah seemed down in the first session, but the standard questions didn't help her feel comfortable. With emotion detection, it taught me to calm her down first, which made her more at ease. It's useful, but the system sometimes lagged

				when generating questions, maybe it needs to be faster.
	Afiqah binti Zulkifli (Fresh Graduate, Human Resource Management)	4	8	I wasn't feeling great in the first session, and the questions didn't notice, which made it feel cold. The second session was better, it asked if I was okay after seeing I was sad, which made me feel understood.
6	Kelvin Ng (Business Development Manager)	6	8	Faris was happy, but the first session's standard questions didn't match his mood, missing a chance to engage him. With emotion detection, it asked about a positive experience, which made him more lively. Overall, it is a good system to help me for interview section.
	Muhammad Faris bin Ismail (Business Administration Student)	4	9	The first session felt flat even though I was happy. The questions didn't reflect my mood. The second session was great, it noticed I was happy and asked about a good moment, which made it more enjoyable. I think the system quite good as it asks more open-ended questions to let me share more freely.

7	Lim Wei Jie (HR Senior)	5	7	Mei Ling seemed neutral in the first session, and the generic questions didn't help me learn much about her. With emotion detection, it asked about her recent work, which got her talking a bit more, but the difference wasn't big. Maybe the system could offer more variety in question types.
	Tan Mei Ling (Fresh Graduate, Marketing)	5	7	The first session's questions were okay but felt standard, and the second session was slightly better—it asked about my recent work after noticing my mood. It didn't make a huge difference for me.
8	Siti Nurhaliza binti Azman (IT Senior)	4	9	Viknesh was nervous in the first session, and the standard questions made it a bit awkward since they didn't help him relax. With emotion detection, it asked if he was okay, which really helped him calm down and share more. The interface could make it better as using different colour for different replies such as emotion detected, questions, explanations.

	Viknesh a/l Suresh (Fresh Graduate, Information Technology)	4	9	I was nervous in the first session, and the questions didn't help me feel better, it felt stiff. The second session was much nicer; it noticed my nervousness and asked if I was alright, which made me feel supported. I think the system could give some tips to relax during the interview.
9	Soh Jia Jia (Media Production Lead)	4	8	In my department, I need someone who is outgoing. In the second session, it helped me to find out that Wei Han is that kind of person which I did not realize in the first session. So, it will help me to find the right person.
	Chong Wei Han (Media Studies Student)	5	8	I was in a good mood in the first session, but the formal question didn't let me show my excitement, it was just okay. The second session was better, it saw I was happy and asked about something I'm proud of, which made it more fun.
10	Cheah Jit Seng (Psychology Team Lead)	5	7	Jia Hui seemed sad in the first session, but the standard questions didn't address her mood, making it harder to connect. With emotion detection, it asked if she was okay,

				which helped her feel more comfortable. It's a good feature, but better to make the questions to be more simplified.
	Wong Shi Hui (Psychology Student)	4	8	The second session is greater for me, the questions asked different between two sessions, it really helps someone when they are feeling nervous. The only problem for me is the questions generated might be a bit long and the questions generated can be faster.

Table 6.2.1 Feedback from Interviewers and Interviewees

6.2.3 Summary of the Result

The results clearly show that the emotion detection feature enhanced the interview experience for both interviewers and interviewees. Without emotion detection, the average ratings were 5.1/10 for interviewers and 4.6/10 for interviewees, with feedback indicating that the generic questions often failed to engage interviewees or adapt to their emotional states, leading to formal and disconnected sessions. For example, interviewees like Desmond Ong and Afiqah binti Zulkifli noted feeling unsupported when their nervousness or sadness went unaddressed, while interviewers like Tan Yong Pang found it challenging to engage neutral candidates without tailored questions.

With the emotion detection model enabled, the average ratings improved significantly to 8.3/10 for interviewers and 8.3/10 for interviewees, reflecting a 63% increase for both groups. Interviewers such as Lee Chun Tat and Ahmad Faiz bin Zainal appreciated how the system identified emotions like nervousness or happiness, allowing them to ask more relevant questions, such as inquiring about proud moments for happy candidates or offering support for those feeling down, which fostered better engagement and connection. Interviewees, including Lee Lei and Viknesh a/l Suresh, reported feeling more understood and supported, with tailored questions making the sessions more enjoyable and less stressful. For instance, Lee Lei rated the second session a 10/10, highlighting how the system's recognition of her happiness and course-relevant questions made the experience fun.

However, participants also provided constructive feedback for improvement. Interviewers like Siti Nurhaliza binti Azman suggested enhancing the interface with color-coded replies for better clarity, while Tan Yong Pang noted inconvenience in typing responses, indicating a need for a more user-friendly input method. Interviewees such as Shalini a/p Ravi and Wong Shi Hui recommended that the system ask more course-specific questions or provide relaxation tips and faster, simpler question generation to further improve the experience. Overall, the A/B testing validates the value of emotion detection in creating a more personalized and engaging interview process, while the recommendations highlight areas for refinement to enhance usability and effectiveness in future iterations of the InterviewAI system.

6.3 Project Challenges

The development of the InterviewAI system faced several significant challenges. One notable difficulty was the system's dependency on the university server, which restricted its operational environment and impacted performance when used in alternative setups. The InterviewAI system relies on the university server's endpoint at /ollama/v1/chat/completions to run the Llama 3.2 Vision model for CV analysis and question generation, requiring a stable and high-speed connection for real-time communication. When attempting to operate the system on a personal laptop, the computational demands, particularly for the Llama 3 model and the CNN-based Emotion Detection Model, resulted in significant delays, consuming a large amount of time due to the lack of dedicated GPU acceleration. For instance, the Llama 3 model's inference on a laptop without a GPU took substantially longer compared to the university server, where GPU support via CUDA and cuDNN enabled faster processing. This limitation not only hindered the system's portability but also its scalability, as it necessitated access to the university infrastructure to maintain the targeted performance levels, posing a challenge for broader deployment.

Another challenge was the real-time emotion detection process, which, despite optimization to capture images only upon detecting emotional changes, still faced issues with accuracy under varying conditions. The Emotion Detection Model, a CNN trained on the RAF-DB dataset, struggled with inconsistent lighting, diverse facial angles, or races, occasionally leading to misclassification of emotions as the model achieving higher than 80 percent accuracy only under ideal conditions. This inconsistency sometimes resulted in inappropriate question adjustments, such as asking a complex technical question when a simpler one was needed due to a mis detected emotion, affecting the Candidate's interview experience.

6.4 Objectives Evaluation

The first objective, to develop an advanced PDF parsing system for comprehensive CV analysis, has been successfully achieved. This was accomplished by implementing a robust parsing mechanism utilizing PyPDF2 to extract structured data from PDF CVs. The system accurately retrieves key information, including the candidate's name, education, skills, work experience, projects, awards, and languages within few seconds, as validated through testing. The extracted data is then processed and categorized, enabling the Llama 3 model to analyze it for generating tailored interview questions that reflect the candidate's professional background and qualifications, thus fulfilling the objective of comprehensive CV analysis.

The second objective, to develop an emotion recognition method to identify the candidates' emotions in real time, has also been successfully achieved. This was accomplished by designing and training a Convolutional Neural Network (CNN) using the RAF-DB dataset, which contains images labeled with emotions such as happiness, sadness, fear, and neutrality. The CNN, structured with four convolutional layers, processes real-time images captured via a webcam to capture candidates' emotion. The model was optimized to capture images only when emotional changes are detected, enhancing efficiency while minimizing resource usage. This real-time emotion recognition enables the system to adapt interview questions dynamically, such as simplifying questions for a nervous candidate, thereby improving the interview experience and meeting the objective of real-time emotion identification.

The third objective, to build a Dynamic Question Generation Framework powered by pretrained LLMs, has been successfully achieved. This was accomplished by leveraging the Llama 3 model, hosted on the university server at the

/ollama/v1/chat/completions endpoint, to generate personalized interview questions based on the Candidate's CV data and detected emotions. The framework constructs detailed prompts incorporating the candidate's background, emotional state, job details, and previous interactions, instructing the Llama 3 model to produce questions across the sections. The system ensures conversational continuity by referencing prior answers in follow-up questions and supports section transitions via the HR Professional's 'next' button, creating an engaging and tailored interview flow that aligns with the Candidate's profile and emotional needs, thus fulfilling the objective of dynamic question generation.

6.5 Concluding Remark

This chapter has provided a comprehensive evaluation of the InterviewAI system, encompassing its testing, performance, challenges, and alignment with the project objectives. Section 6.1 detailed the system's performance metrics, Section 6.2 outlined the A/B testing setup and results, which demonstrated a significant improvement in the interview experience when emotion detection was enabled. Section 6.3 addressed the project challenges, while Section 6.4 evaluated the project objectives, confirming that all were successfully achieved. Collectively, these findings validate the InterviewAI system's potential to revolutionize recruitment by integrating emotion-aware technology, while the identified challenges and participant feedback provide a roadmap for future enhancements.

Chapter 7 Conclusion and Recommendation

7.1 Conclusion

The InterviewAI project was initiated to tackle the inherent challenges of traditional interview methods, such as candidate anxiety and interviewer bias, which often prevent candidates from showcasing their true potential and perpetuate unfair hiring practices. These issues not only affect candidates' performance but also contribute to biased decision-making, reducing workplace diversity and equity. Driven by the goal of creating a more inclusive and empathetic recruitment process, InterviewAI leverages advanced AI technologies to address these problems and enhance the overall interview experience.

The project introduced a innovative solution by integrating emotion detection, resume analysis, and dynamic question generation into a cohesive system. Through real-time emotion detection, InterviewAI adjusts its questions based on the candidate's emotional state, such as offering supportive prompts for nervous candidates or celebratory questions, for those feeling happy, creating a more encouraging and less stressful environment. This adaptive approach helps candidates build confidence gradually, allowing them to perform at their best without the overwhelming pressure typical of conventional interviews. Additionally, the resume analysis model effectively extracts relevant candidate details, while the question generation model produces personalized questions tailored to the candidate's qualifications and experiences, minimizing bias and ensuring a more objective evaluation.

A key strength of InterviewAI lies in its holistic approach, combining multiple AI-driven models to address various aspects of the interview process. By merging emotion detection, resume parsing, and dynamic question generation, the system not only enhances the candidate experience but also supports interviewers in making fairer, more informed decisions based on consistent and unbiased criteria. The A/B testing results, with average ratings improving from 5.1/10 to 8.3/10 for interviewers and 4.6/10 to 8.3/10 for interviewees when emotion detection was enabled, underscore the system's ability to foster greater engagement and comfort, validating its effectiveness in real-world scenarios.

In conclusion, InterviewAI marks a significant step forward in recruitment technology by delivering an efficient, emotionally intelligent, and equitable AI-driven solution. By overcoming the limitations of traditional interviews, this project paves the way for a more effective and insightful hiring process, benefiting both candidates and employers. While challenges such as PDF-only parsing and server dependency were identified, the system's achievements lay a strong foundation for future advancements, with the potential to transform recruitment into a more equitable and empathetic process

7.2 Recommendation

The following recommendations address key areas for improvement in the InterviewAI system, focusing on diversifying question types with additional sections, adding support features, and optimizing system portability and performance. These enhancements aim to improve the system's adaptability, user experience, and scalability, while acknowledging budget constraints that impacted the feasibility of some solutions during the project.

Optimize System Portability and Performance

The InterviewAI system's dependency on the university server, requiring GPU acceleration for efficient processing, significantly limited its portability. Personal laptops without GPUs experienced notable delays, making the system less suitable for deployment outside controlled environments like a university, where such infrastructure is readily available. To address this, the system should be optimized for broader accessibility by leveraging cloud-based solutions. Hosting the Llama 3 model on a cloud platform like AWS, Google Cloud, or Microsoft Azure would allow the system to utilize scalable GPU resources, ensuring real-time performance without requiring local GPU hardware. This cloud deployment would also enable remote access, making the system usable in diverse settings such as corporate offices or remote interviews, eliminating the need for specialized university infrastructure. However, I considered this solution during the project, but budget constraints prevented its implementation, as cloud services like AWS or Azure involve recurring costs that were beyond the project's allocated funds. If the budget allows in future iterations, cloud hosting would be an ideal solution to ensure portability and performance.

Diversify Question Types and Add Support Features

The A/B testing feedback revealed that while the InterviewAI system's current four section, Initial, Personal, Life, and Technical, cover key areas like background, experiences, and skills, some participants felt the questions could be broader to assess other aspects of a candidate's fit. For further improvement, the Question Generation Model should be expanded to include additional sections, such as a Cultural Fit section, which is not currently part of the system. This section would help interviewers evaluate how well a candidate's values, work style, and interpersonal approach align with the company's culture through questions like "What kind of work environment do you thrive in, and how does that match our company's collaborative culture?" or "How do you handle feedback, considering our emphasis on open communication?" By adding this section, interviewers can better determine if the candidate is suitable for the company based on their responses, ensuring a better long-term fit. Additionally, the system can leverage AI to assist in evaluating responses by implementing a rating mechanism that scores answers based on alignment with the company's cultural values. For example, assigning a score from 1 to 5 for responses to cultural fit questions, where a higher score indicates better alignment with predefined cultural traits like teamwork or innovation. This AI-driven rating, combined with the interviewer's judgment, would provide a more objective and consistent assessment, helping to identify the most suitable candidates for the organization.

Implement Continuous Monitoring and Updates

To keep the InterviewAI system effective and aligned with the evolving demands of recruitment, a comprehensive framework for continuous monitoring and updates is vital. A feedback mechanism should be embedded in the interface, enabling interviewers and candidates to provide comments after each session, highlighting areas for improvement, such as irrelevant questions or inaccurate emotion detection, allowing developers to quickly address these concerns. Additionally, the Llama 3 and Emotion Detection Models should be regularly updated, ideally on a quarterly basis with fresh datasets reflecting current trends, including new CV formats like visual resumes or LinkedIn profiles, shifting emotional expression patterns influenced by cultural changes or virtual interview settings, and industry-specific terms relevant to sectors like technology, healthcare, or marketing, ensuring consistent data extraction and improved emotion detection accuracy across diverse scenarios. An analytics dashboard should also be introduced to monitor key metrics, such as question, emotion detection accuracy, and average session duration, providing actionable insights for ongoing enhancements. By systematically gathering and analyzing user feedback alongside performance data, the development team can prioritize updates that enhance user experience and system performance, ensuring the InterviewAI system remains accurate, relevant, and adaptable to the dynamic recruitment landscape.

REFERENCES

[1] A. Dubey et al., "The LLaMA 3 Herd of Models," *arXiv preprint arXiv:2407.21783,* 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[2] A. Jaiswal, A. Raju, and S. Deb, "Facial Emotion Detection Using Deep Learning," Jun. 2020.

[3] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," *arXiv preprint arXiv:1906.02243,* Jun. 2019. [Online]. Available: https://arxiv.org/abs/1906.02243

[4] J. Daniel and J. Martin, "Speech and Language Processing Chatbots & Dialogue Systems," Aug. 2024. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/15.pdf

[5] J. Zhang et al., "Advances in Multimodal Large Language Models," *arXiv preprint arXiv:2403.12475*, 2024. [Online]. Available: https://arxiv.org/abs/2403.12475

[6] L. F. C. Quiroga et al., "Enhancing Recruitment Processes with Multimodal Large Language Models," *Future Internet*, vol. 16, no. 7, pp. 219, 2024, doi: 10.3390/fi16070219.

[7] M. I. Georgescu, R. T. Ionescu, and M. Popescu, "Local learning with deep and handcrafted features for facial expression recognition," *IEEE Access*, vol. 7, pp. 64844–64858, 2019, doi: 10.1109/ACCESS.2019.2917266.

[8] M. Ozdemir, B. Elagoz, Aysegulalaybeyoglu, R. Sadighzadeh, and A. Akan, "Real Time Emotion Recognition from Facial Expressions Using CNN Architecture."

[9] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Trans. Affective Comput.*, vol. 10, no. 1, pp. 18–31, 2017, doi: 10.1109/TAFFC.2017.2740923.

[10] P. Anantharaman, S. Cheung, N. Boorman, and M. Locasto, "A Format-

[11] P. Paudel, S. Khadka, and R. Shah, "Optimizing Nepali PDF Extraction: A Comparative Study of Parser and OCR Technologies," *arXiv preprint arXiv:2407.15494,* Jul. 2024. [Online]. Available: https://arxiv.org/abs/2407.15494

[12] Paradox.ai, "AI-Powered Hiring Solutions," 2025. [Online]. Available: https://www.paradox.ai/

[13] R. C. Schank, "What Is AI, Anyway?," *AI Magazine*, vol. 8, no. 4, pp. 59–59, 2019, doi: 10.1609/aimag.v8i4.623.

[14] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Trans. Affective Comput.*, vol. 13, no. 1, pp. 119–137, 2020, doi: 10.1109/TAFFC.2020.2981446.

[15] S. Minaee et al., "Large Language Models: A Survey," *arXiv preprint arXiv:2402.06196*, Feb. 2024. [Online]. Available: https://arxiv.org/pdf/2402.06196

[16] T. Brown et al., "Language Models are Few-Shot Learners," *arXiv preprint arXiv:2005.14165*, Jul. 2020. [Online]. Available: https://arxiv.org/pdf/2005.14165

[17] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint arXiv:1907.11692*, 2019. [Online]. Available: https://arxiv.org/pdf/1907.11692

APPENDICES

A.1 Dataset

Partially dataset shown for Emotion Detection (angry)

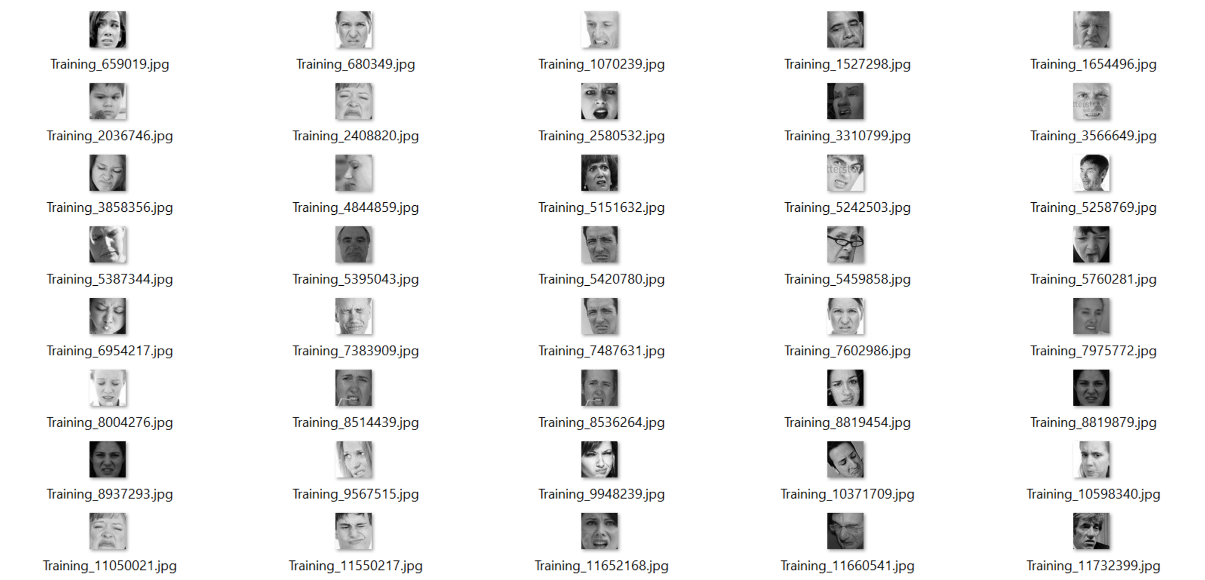
Partially dataset shown for Emotion Detection (happy)

Partially dataset shown for Emotion Detection (fear)

Partially dataset shown for Emotion Detection (neutral)

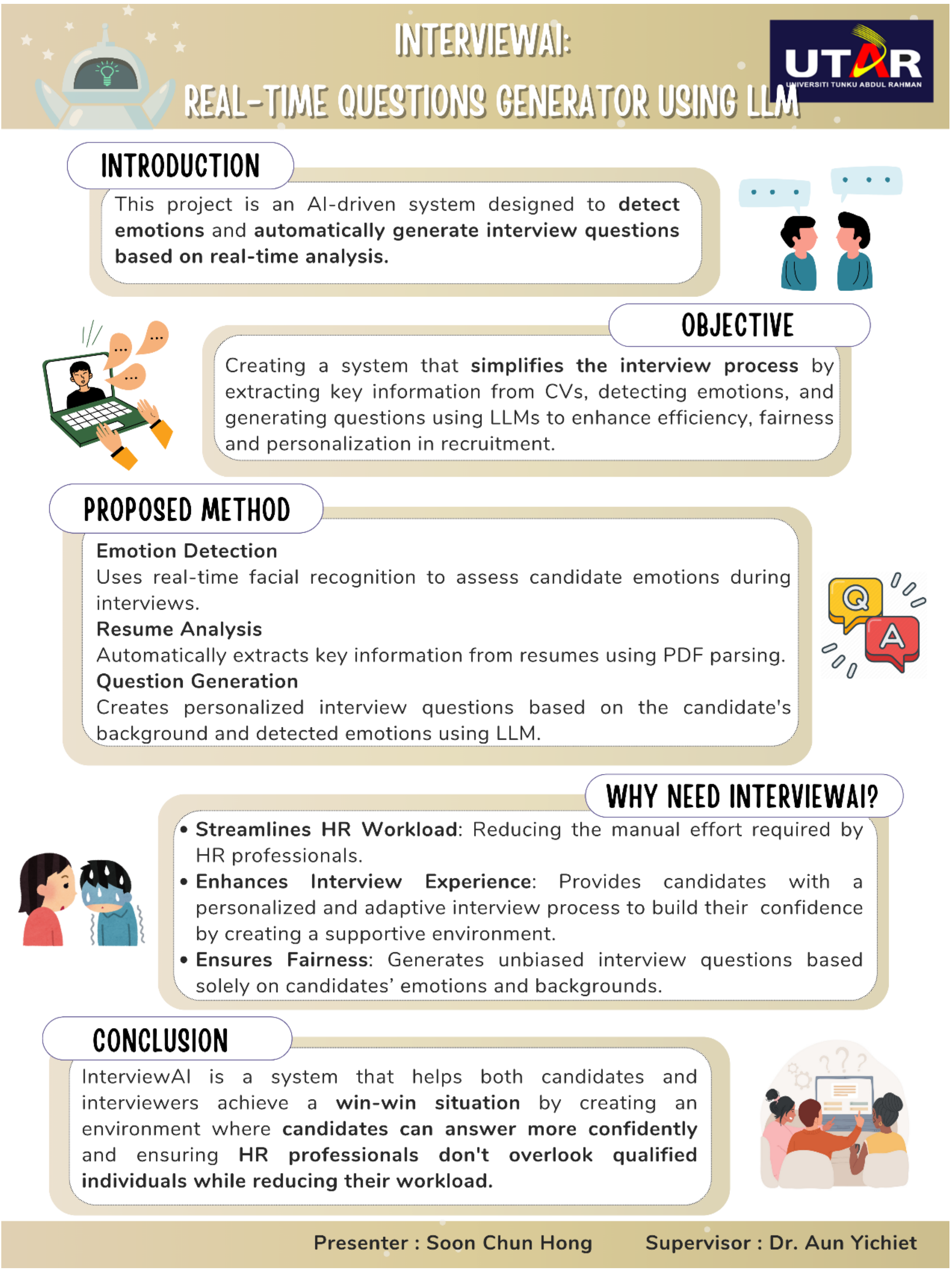
Partially dataset shown for Emotion Detection (sad)

Partially dataset shown for Emotion Detection (surprise)



Partially dataset shown for Emotion Detection (disgust)

A-2 Poster


INTERVIEWAI:
REAL-TIME QUESTIONS GENERATOR USING LLM
UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
INTRODUCTION
This project is an AI-driven system designed to detect emotions and automatically generate interview questions based on real-time analysis.
OBJECTIVE
Creating a system that simplifies the interview process by extracting key information from CVs, detecting emotions, and generating questions using LLMs to enhance efficiency, fairness and personalization in recruitment.
PROPOSED METHOD
Emotion Detection
Uses real-time facial recognition to assess candidate emotions during interviews.
Resume Analysis
Automatically extracts key information from resumes using PDF parsing.
Question Generation
Creates personalized interview questions based on the candidate's background and detected emotions using LLM.
Q
A
WHY NEED INTERVIEWAI?
• Streamlines HR Workload: Reducing the manual effort required by HR professionals.
• Enhances Interview Experience: Provides candidates with a personalized and adaptive interview process to build their confidence by creating a supportive environment.
• Ensures Fairness: Generates unbiased interview questions based solely on candidates' emotions and backgrounds.
CONCLUSION
InterviewAI is a system that helps both candidates and interviewers achieve a win-win situation by creating an environment where candidates can answer more confidently and ensuring HR professionals don't overlook qualified individuals while reducing their workload.
Presenter : Soon Chun Hong
Supervisor : Dr. Aun Yichiet